

**ADVANCE DYNAMIC WEB BASED HOSPITAL
DATABASE MANAGEMENT SYSTEM**

BY

BABATUNDE HABEEBULLAH FEMI

G2017/MSC/CITE/006

A DISSERTATION REPORT

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
AWARD OF THE DEGREE OF MASTERS OF SCIENCE (MSc) IN
INFORMATION AND TELECOMMUNICATION ENGINEERING**

IN THE

**CENTRE FOR INFORMATION AND TELECOMMUNICATION
ENGINEERING (CITE), FACULTY OF ENGINEERING, UNIVERSITY
OF PORTHARCOURT, NIGERIA.**

DECEMBER, 2018.

DECLARATION

I hereby declare that this thesis work is mine and that it has not been submitted anywhere for the purpose of awarding a degree to the best of my knowledge.

BABATUNDE HABEEBULLAH FEMI

G2017/MSC/CITE/006

Signature

.....

Date

CERTIFICATION

This is to certify that this dissertation titled “**ADVANCE DYNAMIC WEBBASED HOSPITAL DATABASE MANAGEMENT SYSTEM**” is an authentic work done by me under the supervision of Dr. J. J. Biebuma for the partial fulfillment of the award of Masters of Science Degree (MSc) in Information and Telecommunication Engineering at the University of Port Harcourt.

Dr. J. J. Biebuma
-------------------	-------	-------

Dissertation Supervisor	Signature	Date
--------------------------------	------------------	-------------

Dr. Bourdillon Omijeh
-----------------------	-------	-------

Director CITE	Signature	Date
----------------------	------------------	-------------

--	-------	-------

External Supervisor	Signature	Date
----------------------------	------------------	-------------

--	-------	-------

Chairman Board Examiner	Signature	Date
--------------------------------	------------------	-------------

DEDICATION

This work is dedicated to Almighty God who in his infinite mercy counts me as a living soul and protected me through the hurdles I faced in the course of this program.

ACKNOWLEDGEMENT

Special thanks to CITE Director Engr. (Dr.) Omijeh Bourdillon, my Dissertation Supervisors Dr. J. J. Biebuma and Mr. Adewale Samuel for their guidance and input on this dissertation work.

To all CITE staff (Engr. Dr. Dan, Mrs. M. Peter, Mr. Emmanuel etc.), the industry lecturers (Tobechukwu Obeifuna - MTN, Evans Ibara – MTN, Solomon Aremu – Nokia, Alex – Nokia etc.), my colleagues (Ilekoya Femi, Mohammad Sadiq, Agughalam Davis, Onu Praise and Mumuni Mofehintolu), I say thanks to you all for the supports.

With joy, I say a big thank you to my parents Hon. Mr. and Mrs. Babatunde, my siblings, Yussuf Mustapha, Bello Muhammed, Okafor Chinenye, Grace Purity and all the loved ones. You are the best.

ABSTRACT

Hospital Database Management System (HDBMS) is designed to maintain the data of hospital staff, patients, medical records, treatments and infrastructures in the hospital. In this modern world where every routine has been digitalized to hasten processes and give better productivity, HDBMS has also been transitioned from the use of paper repository to the use of computer and this is better done on computer with the use of application/software that gives the same function as expected of a hospital management system. The Dynamic HDBMS application is structured into modules and each of the modules have their functions in the HDBMS.

TABLE OF CONTENTS

	<i>Pages</i>
<i>Title Page</i>	
<i>Declaration</i>	<i>1</i>
<i>Certification</i>	<i>ii</i>
<i>Dedication</i>	<i>iii</i>
<i>Acknowledgement</i>	<i>iv</i>
<i>Abstract</i>	<i>v</i>
<i>Table of Contents</i>	<i>vi</i>
<i>List of Figures</i>	<i>ix</i>

CHAPTER 1: INTRODUCTION **1**

1.1	Background Study	1
1.2	Statement of Problem	2
1.3	Dynamic HDBMS	3
1.4	Significance of Work	3
1.5	Scope of Work	4

CHAPTER 2: LITERATURE REVIEW **5**

2.1	Limitation of the Existing Work	5
2.2	Related Work	6
2.3	Big Data Analytics	12
2.4	Key Performance Index	12
2.5	Data Privacy and Security	13
2.6	Algorithm	14

CHAPTER 3: METHODOLOGY **15**

3.1	System Design	15
3.2	Methodology	17
3.3	Architectural Plan (Normal System)	18
3.4	Architectural Plan (Emergency/Operation and Child Delivery)	20
3.5	System Flowchart	21
3.6	Design Algorithm	22
3.7	Input Design Specification	24
3.8	Hardware and Operating System Requirement	24
3.8.1	Hardware Specifications	25
3.8.2	Software Specification	26

CHAPTER 4: RESULT AND DISCUSSION **27**

4.1	Subscriber Login Form	27
4.2	Login Form	28
4.3	Staff Login Panel	29
4.4	Registration Form	30
4.5	Consultation Form	34
4.6	Nurse Form	37
4.7	Laboratory Form	38
4.8	Patient Form	40
4.9	Billing Form and Sales form	42
4.10	Stock Form	44
4.11	Reference Form	45

CHAPTER 5: CONCLUSION AND RECOMMENDATION	49
5.1 Conclusion	48
5.2 Recommendation	48
5.1 Contribution to Knowledge	48
REFERENCES	49
APPENDIX I	50
APPENDIX I1	61

LIST OF FIGURES

	<i>Pages</i>
Fig 2.1 Challenges Faced Using HDBMS	5
Fig 2.2 Conceptual model for Patient Database	7
Fig 2.3 Block Diagram on Online Management System	9
Fig 2.4 Hierarchical model in designing Management System	10
Fig 2.5 Network model in designing Management System	10
Fig 2.6 Hospital Queue System Use Case Diagram	11
Fig 3.1 Architectural Plan (Normal System)	19
Fig 3.2 Architectural Plan (Emergency/Operation and Child delivery)	20
Fig 3.3 System Flowchart	21
Fig 4.1 Subscriber Login Form (Source: Appendix 1, Login Form)	27
Fig 4.2 Login Form (Source: Appendix 1, Login Form)	28
Fig 4.3 Staff Login Form (Source: Appendix 1, Login Form)	29
Fig 4.4.1 Staff Registration Form (Source: Appendix 1, Registration)	30
Fig 4.4.2 Staff Registration Form (Source: Appendix 1, Registration)	32
Fig 4.4.3 Subscriber Form (Source: Appendix 1, Registration)	33
Fig 4.4.4 Patient Registration Form (Source: Appendix 1, Registration)	33
Fig 4.5.1 Consultation Form (Source: Appendix 1, Consultation Form)	34
Fig 4.5.2 Consultation Form (Source: Appendix 1, Message Module)	35
Fig 4.5.3 Consultation Form (Source: Appendix 1, Message Module)	36
Fig 4.6.1 Vital Signs (Source: Appendix 1, Vital Signs)	37
Fig 4.6.2 Vital Signs (Source: Appendix 1, Vital Signs)	38
Fig 4.7 Laboratory (Source: Appendix 1, Laboratory Form)	39
Fig 4.8.1 Patient form (Source: Appendix 1, Patient Form)	40

Fig 4.8.2	Patient form (Source: Appendix 1, Patient Form)	41
Fig 4.9.1	Billing Form (Source: Appendix 1, Billing Form)	42
Fig 4.9.2	Billing Form (Source: Appendix 1, Billing Form)	43
Fig 4.9.3	Sales Form (Source: Appendix 1, Sales Form)	44
Fig 4.10	Stock Form (Source: Appendix 1, Stock Form)	45
Fig 4.11.1	Reference Form (Source: Appendix 1, Reference Form)	46
Fig 4.11.2	Reference Form (Source: Appendix 1, Reference Form)	46
Fig 4.11.3	Reference Form (Source: Appendix 1, Reference Form)	47
Fig 4.11.4	Reference Form (Source: Appendix 1, Reference Form)	47
Fig 7.1	Letter of Introduction	61

CHAPTER 1

INTRODUCTION

1.1 Background Study

HDBMS started as a form of Hospital Information System (HIS) which its main function is to manage the hospital activities such as administrative, financial, legal, infrastructural and medical operations. The overall goal of the information management function is to obtain, manage, and use information to improve health care and medical services, performance, governance and management and support processes (Nkanata, 2015). HDBMS started as a paper documentation before it was revolutionized to the use of spreadsheet that makes the documentation easier, it is of no doubt that this two forms of documentation takes a longer time and data retrieval from the database may be difficult after a long period. The whole spreadsheet method was applied after the invention of computer to make excel sheet which even make the process faster compared to spreadsheet documentation. But in order to extend the simple function of HDBMS, an application / software is needed that will give access to various data input and will also make access to the inputted data easy. This is what bring rise to HDBMS software.

The HDBMS software is of two types namely; the hardcoded HDBMS and the dynamic HDBMS. The later HDBMS software has some information already built on the software and these information always serve as the factor of operation of the HDBMS while the dynamic HDBMS work as just hardcoded HDBMS but it allows input and can be easily updated with new factors of

operation unlike the hardcoded that there is no privilege of new update unless the whole code is debugged.

1.2 Statement of Problem

The ideal way of operation seems smooth and stress free but it would have been more perfect if HDBMS can handle more of the operation management in other to have an executive and a more professional system. Here are the things the ideal system lacks;

1. Staff/Doctor active communication via messaging platform
2. Active analysis page that can be published such as mortality rate, infant obesity rate etc.,
3. Log file to track who is active to take up a test or patient treatment,
4. Especially in Nigeria, patient lack medical record/history,
5. Payment processes method which requires cashier, bursar, accountants and auditors.

These features mentioned above may seem so negligible but indeed they go a long way if the health care sector or the health information system is to be improved on. The advantages of the additional features to common HDBMS used in hospital is more when compared to the money and time spent on managing the existing common HDBMS in hospitals. Also, creating an active communication platform requires both the local network messaging and email integration, the email integration requires an external service such asetc. Also, thus active communication helps the hospital to keep in contact with their patient as they sends messages on updates in medical record and some other latest development that will interest them to come back for their services.

1.3 Dynamic HDBMS

To make an HDBMS that is advanced version of the common HDBMS, the aim of this project is to build dynamic HDBMS software that allows easy inputting, updating and managing of database of hospital records. This system will have features that facilitate the work of the human resource department, account department, doctors, lab attendant, radiologist, therapist, pharmacist and other staffs. To make all the proposed features work as expected here are the functions to be added;

1. A general platform for all hospitals in the country: the platform will run the management system for the hospitals that are registered on it.
2. Messaging module that ensure active communication between the staffs and analysis page that gives basic statistics of the hospital.
3. A consultation page with detailed treatment suggestion and patient record.

1.4 Significance of Work

Advantages of HDBMS are as follow;

1. It facilitates consultation process of patients in the hospital as database holds the record of the entire registered patient.
2. It enhances management of staff and hospital facilities and as well keeps track of status of infrastructure in hospital.
3. Medical records alongside the staff activities are managed using HDBMS i.e. human resources.
4. Financial records are easily remitted and audited using database management system.

1.5 Scope of Work

To achieve the aims and objectives mentioned earlier the need of a programming language is essential in building software or application, because of windows performance and compatibility; it is preferred to use C# programming language.

The whole system is to be divided into functional module that span from;

1. Normal HDBMS processes module
2. Inpatient module
3. Medication module
4. Pharmaceutical, stock and inventory module
5. Human resources module
6. Messaging module
7. Payment module
8. Security module

After these module had been designed and built, all the modules are to be integrated together to work as the HDBMS system and afterward it would be launched to be a web based application.

CHAPTER 2

LITERATURE REVIEW

2.1 Limitation of the Existing Work

In 2015, Nkanata researched on the use of hospital management system and comparative studies of usage of the system between public and private hospitals; this was based on basic characteristics of information management system which are completeness of data, accuracy, timeliness, and relevancy of the data. To achieve this education distribution of workers in the hospital were taken and was found that about 50% of workers in both public and private hospitals are diploma holders, introduction of information management system has improved the quality of service of private hospitals by 74% and assumed to have improved public hospital by 26.83%, financial imperatives and accuracy of data has improved by about 80% in both hospitals. The hospital information management system used was 49% user friendly, 18% availability of the system and 33% quick access to patient data.

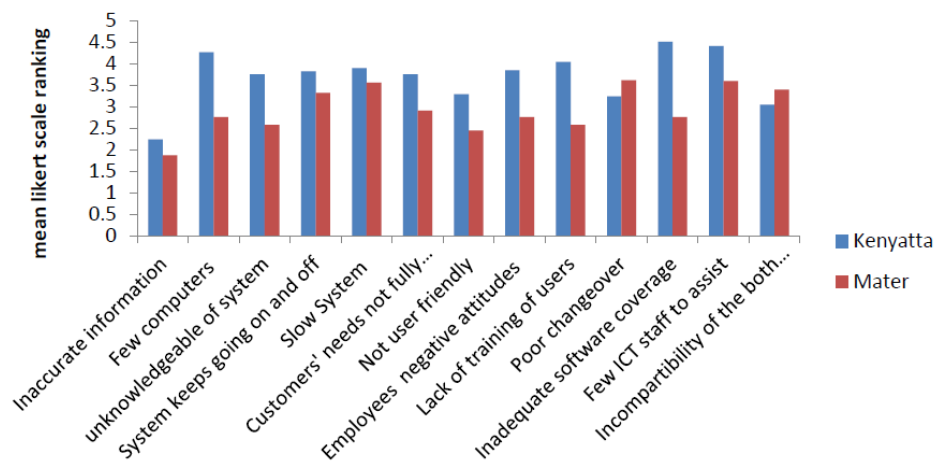


Fig 2.1 Challenges Faced Using HDBMS

2.2 Related Work

In a related application designed for doctors and pathologists by Gunjan Yadav et'al, the application is of three modules namely; login and update module, doctors and Patients module, Test details and update patient record module. So, these three modules are further divided into panel such that the login and update module is classified as admin panel i.e. only the admin can have access, and doctors and Patients module classified under Hospital panel i.e. doctors will have access to this panel, and lastly the patient record module is classified as a lab panel i.e doctors and pathologist will have access. Digvijay H. et'al pointed out the limitations in the existing versions of HDBMS and these limitations include ability to take more users (more than 20) at a time, communication gap between the employees and the management and a poor rewarding system. In the design of the HDBMS, they deploy an architectural scheme which is shown in the diagram below;

With this architectural design, the HDBMS is designed to increase the response time and reduce the complexity accounting section passes through on financial records.

Adebisi O.A et'al, 2015 research on design and implementation of hospital management system by using Hypertext Preprocessor (PHP) as the backend language with My Structured Query Language (MySQL) for database storage and Hypertext Markup Language (HTML) and Cascading Style Sheet (CSS) for the front end. The purpose of this research was to determine the current state of management procedures in one of the hospitals in Nigeria and how it can be improved, the weaknesses of the current system according to the authors are;

- most hospital staff finds it tiresome computing patient data,

- hospital record files are still in use in the system which is susceptible to security problem,
- retrieving data take long time,

And some of these challenges were solved with software and hardware requirement of devices in the hospital, also data verification is put in place to confirm the data been put into the database. While the other solutions come from the design and the architecture of the implemented proposed system.

Nadeem Mahmood,2012 researched on the data and knowledge management in designing healthcare information systems, the health care industry is categorized as a knowledge-based community due to the sharing of knowledge that exist between hospitals, physicians, patients, laboratories, pharmaceuticals, clinics, pharmacies and customers. This research was done to know how data and knowledge can relate in designing a fully flexible healthcare management system. The patient-doctor entity relationship has enabled some other systems that facilitate in giving a good health services in other to provide information sharing.

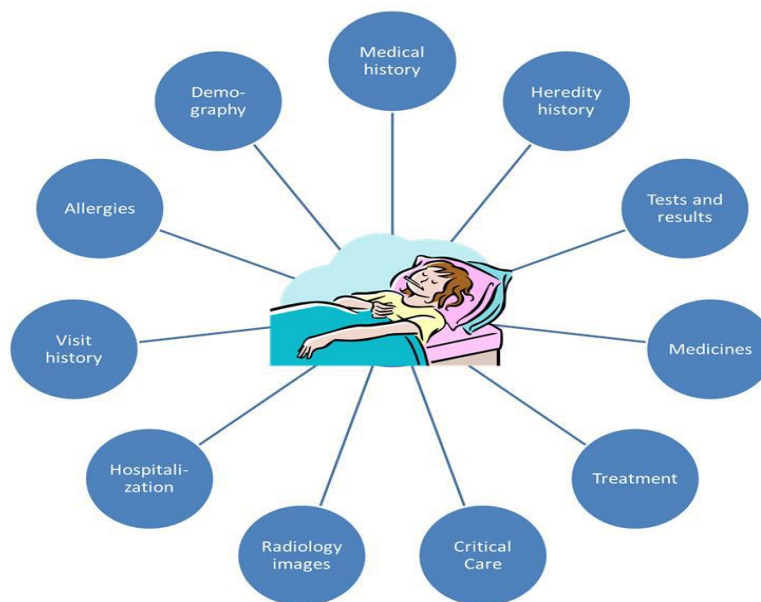


Fig 2.2 Conceptual model for Patient Database

Data and knowledge based management system design includes health workflow management and ICT, E-health (data collection, analysis, sharing of information) and M-health (combination of wireless technologies and E-health). The common features in these three designs are ability to access information by doctors, patients, researchers, health care professionals and other in order to get desired changes in the information system and also these designs are less robust, expensive while providing more realistic method for effectiveness of alternative clinical practices.

Database optimizing services is a very essential factor needed to be considered in designing a management system Andrian Ghencea and Immo Gieger, 2010 gave an analysis on database optimization services and was concluded in the research that high quality security and availability is what is entailed in an optimized system. Database management system been used as a research tools in most sectors is essential tool that is needed in the healthcare sector, the relational database may incur cost if it lacks optimization due to the millions of data transferred per second, this make optimization a key research domain. There are different ways in which database optimization is facilitated, these includes indexing; increasing number of queries on the database which can vary from one another. Indexing is of two types namely; the clustered which affect the physical order of the table arrangement and the non-clustered which does not affect the arrangement of the database table. Another way of optimizing database is using the GREEDY-SEQ algorithm which relates the number of configurations to the number candidate structure, the limit in this type of optimization is when there is large number of candidate structure.

Mansi Chitkara et'al describes the feasibility study required in designing a standard hospital database management system and this study is based on these questions;

- The user's demonstrable needs and how candidate meets them,
- Resources available for candidate system,

- The impacts of candidate system on the hospital
- Is it worth to solve the problem?

The steps taken are in four categories which are; technical feasibility, economic feasibility, operational feasibility, schedule feasibility. These feasibility steps help in preparing a good flowchart, enumerating proposed solution to the system and helps in measuring performance and cost data.

In the design and development of online hospital management information system, 2013 by Harpreet Kaur, this online management system is built to have better speed and performance than the casual or traditional hospital management system and also the system comprises of four modules namely; patients detail, billing and login. The flow of the system is shown below;

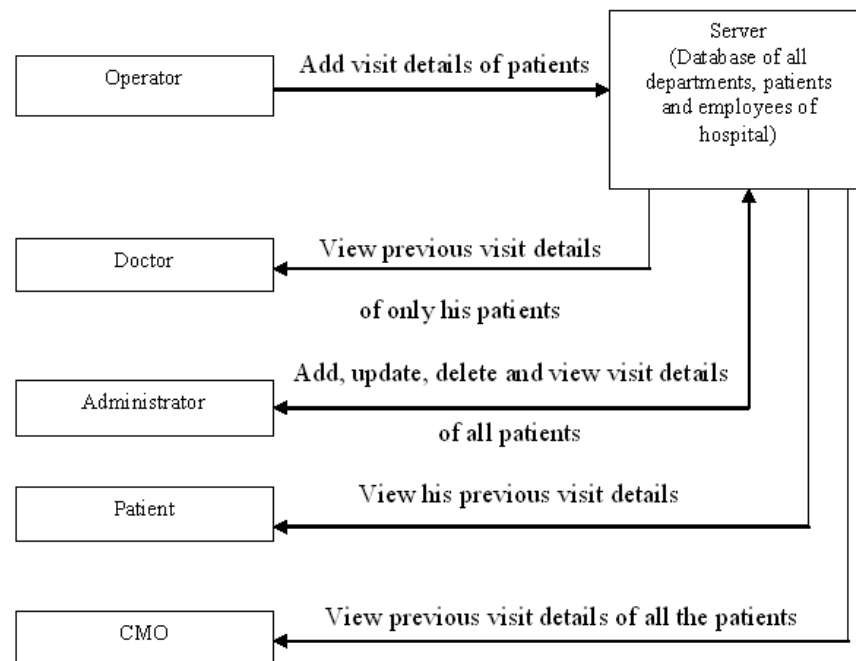


Fig 2.3 Block Diagram on Online Management System

In the research made by Sweta singh, 2015 to highlight the specific properties of database management system which was characterized as efficient data access, data integrity and security, data administration, concurrent access and crash recovery, reduced application development time and these properties were valued with the types of database management system by giving a model for each;

Hierarchical Model

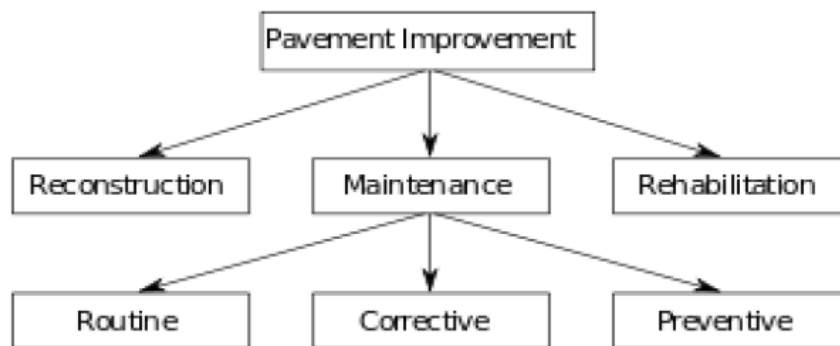


Fig 2.4 Hierarchical model in designing Management System

Network Model

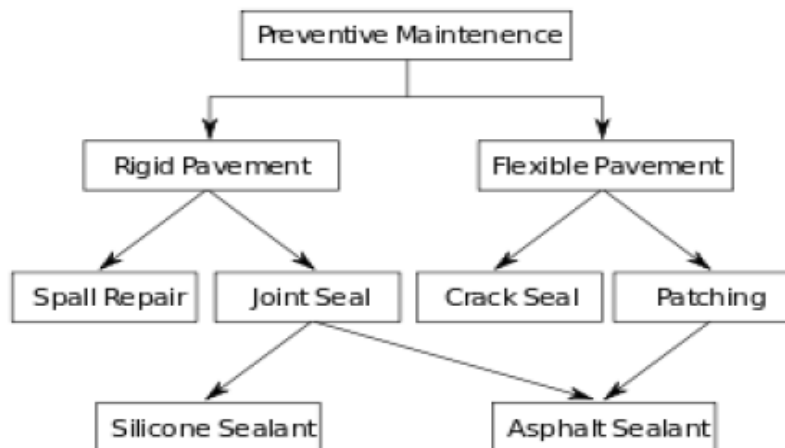


Fig 2.5 Network model in designing Management System

Hospital service queue management system designed with wireless approach in 2015 by Manoon Ngorsed et'al, waiting for any service is counterproductive especially in the health sector as it consumes most of the valuable time that is enough to engage in some other things so there is a need of a queue management that helps reduce the inconvenience to the public. The design is made such that patients and costumers can remotely access their queue over the internet, the system boundary and architecture takes four entities;

- system admin: represents administrator that grants access to all system features,
- queue admin: represents hospital queue administrator,
- queue operator: represents the person that takes care of each queue,
- queue client: represent the person that requires hospital service.

The design is made as shown below;

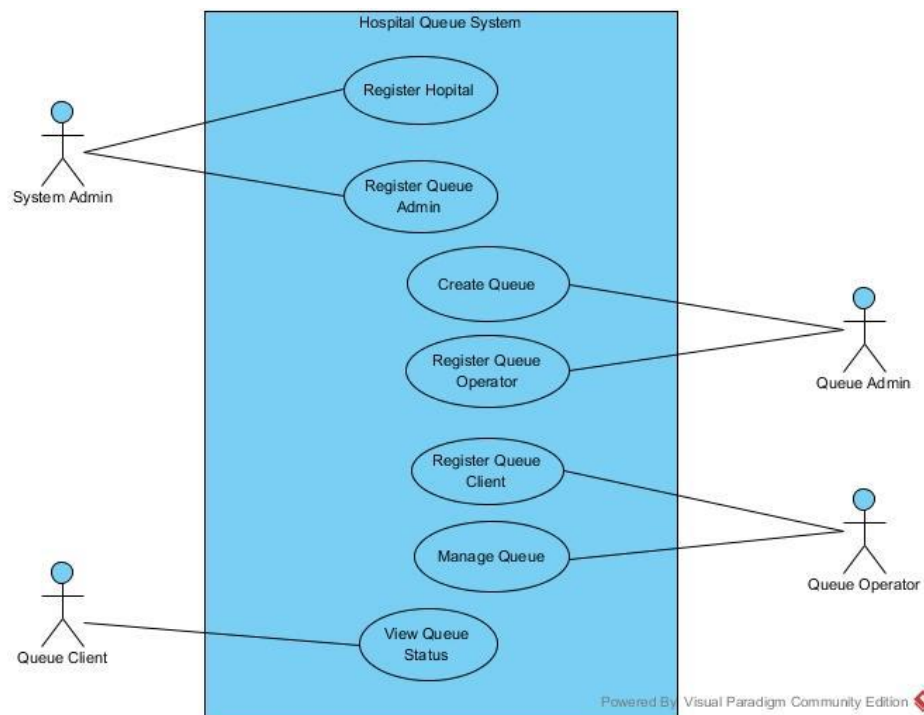


Fig 2.6 Hospital Queue System Use Case Diagram

The design was implemented using queue management mechanism and system prototype implementation.

In international journal of database management systems (IJDMS) which received contributions from notable universities across the world in 2016 focused on various topics related to database management which includes constraint modeling and processing, data and information integration & modeling, data semantics and ontologies, data mining, expert systems and interoperability, very large databases, query processing and optimization, metadata management etc. These topics show how important application of database is essential.

2.3 Big Data Analytics

Big data analytics is another very important area of concentration when designing a fully flexible database management, though this aspect is most often dodged by most of the developers and database expert because of cost and security issues. Sanskruti Patel et'al, 2016 researched on big data revolution in health care sector and give the opportunities and challenges in operating big data in health care sectors. These opportunities includes; it brings about innovations, reduce healthcare cost, it help strengthens the preventive care and identification and tracking of patients, while the challenges includes data integration, privacy and security concerns, requires experts knowledge.

2.4 Key Performance Index (KPI)

Mike Suave emphasized on the Key Performance Index (KPI) to be used by all hospitals to determine the quality management level of operation, the KPI are

categorized as the integral part or tools for analyzing the data collected in the hospital and this can later be used to form the balance act. The KPI hospitals should consider to know the effectiveness of their management system includes; patient satisfaction, incidents, death rate, medication error, admission/discharge wait time, admission rate, readmission rate and patient referrals. So In designing a management system for hospital, these KPIs are needed to be considered and carefully integrated as these metric is what determines the success of hospital management system.

2.5 Data Privacy and Security

Research on medical ontology for using information technology in health sector put patients as the primary importance such that all the activities in the hospital that span from monitoring, treatment, diagnosis, reporting, findings, billing, affiliation and identification revolve round the patient key, this medical ontology was used to design a conceptual patient database. Further research on the medical ontology emphasized the security needed for patient database integration on platform such as ONIONS (TOR) platform, Telemedicine that requires ICT infrastructure and services such as healthcare service providers, government agencies etc. The conclusion of this research by Nadeem Mohmood et'al, 2012) was the basic requirement for an effective hospital information system which includes provision of a good statistical tool for correct analysis of data, making the management system less expensive, robust and more realistic which must be easy to use.

Hospital Management System (HMS) and Hospital Information System (HIS) follow a strict standard in guarding the information on the database, one of the

standards use in protecting the privacy of data is Healthcare Insurance Portability and Accountability Act (HIPAA) privacy guidelines for medical record. The privacy rule referred to as "protected health information (PHI) must be complied by all hospitals holding individually identifiable health information in any form on paper or electronics. Premkumar Balaraman et'al, 2013 further measure hospital performance with the five principles recommended by WHO, these five principles were used to build the modules of an HDBMS.

2.6 Algorithm

There has been many popular algorithm in use to improve the health care sector such as health scores for quantifying illness, medical algorithm for better decision aids and safer care, Fourier transform algorithm to enhance senses and helps to keep the system dynamic, encryption algorithm for improving the security of the system, MUMPS that serves as health care's operating system, data matching to help search easier, BLAST algorithm for analyzing gene and protein sequences etc. Each of these algorithms has contributed a lot to the services of HDBMS overtime, apart from these the general algorithm use requires first registration of staff and patient on the HDBMS while access is giving to customer care/receptionist, doctors, pharmacist and lab attendant to put their records.

CHAPTER 3

METHODOLOGY

3.1 System Design

The HDBMS is designed to operate and function for nearly all the staff in the hospital especially the doctors and consultants, Nurses, Lab attendant, Human resource managers and financial Officers.

It is built such that when the application is installed in the hospital system at first time, it requests for the subscription ID which is to be obtained from the controlling body and afterward the subscription ID is input into the system it prompts the main login page which one of the hospital admins will have login details, the system automatically recognizes login detail as an admin detail and redirects it to the admin page where admin roles are highlighted. From the admin page the admin can register staff and also have access to some services, after registering the staff, the individual staff can now proceed from the login page of the HDBMS and login. The login function checks the database for the user profession and directs them to the appropriate page to carry out the daily service.

Another section of the system is the patient side, when patient visits a hospital the customer care/receptionist attends to the patient for service like buying of card and accepts request for consultation. If the patient is new and doesn't have a card yet the customer care/receptionist register the patient into the database but if the patient already have a card, the receptionist checks if it is on appointment checking the database for patient ID that matches the patient own. The consultation process continues as customer care/receptionist directs the patient. To the nurse available for vital signs test, the doctor receives the patient afterward for consultation; the

doctor's page allows the doctor to check the patient past medical records while notifying all various departments of services to offer the patient on board. All the data collected from above flows are stored in the database and are accessible by the doctors to make necessary health decisions for their patients.

Another module of the HDBMS designed is financial record tracking of the patients and fund monitoring of the hospital. It flow starts directly from the patient making payment for consultation or buying of card to payment for the prescriptions and lab tests made by the doctor, The billing records for the services and stocks collected by patients are transferred to their respective profile page, from the profile page financial officers can details of what the patient is to pay for and change the status of payment of the patient to paid or pending.

Also, the pay roll is handled by the human resource to make payment of all the staff salaries and allowances, and also the record for stock orders and sales is kept by the pharmacist. In the process, it would be easy for the hospital to know their income and expenditure at any time interval that is needed.

In order to effectively utilize big data and prevent overload and dragging of the system, Image and document files are saved inside a folder of the system instead of on the database, with this the memory can be efficiently managed and be prevented from memory consumption that happens when files are called by a function of the system.

The data security aspect was executed by salting algorithm in which suffixes and prefixes are added to the passwords set by the admin or before it is been processed and put into the database.

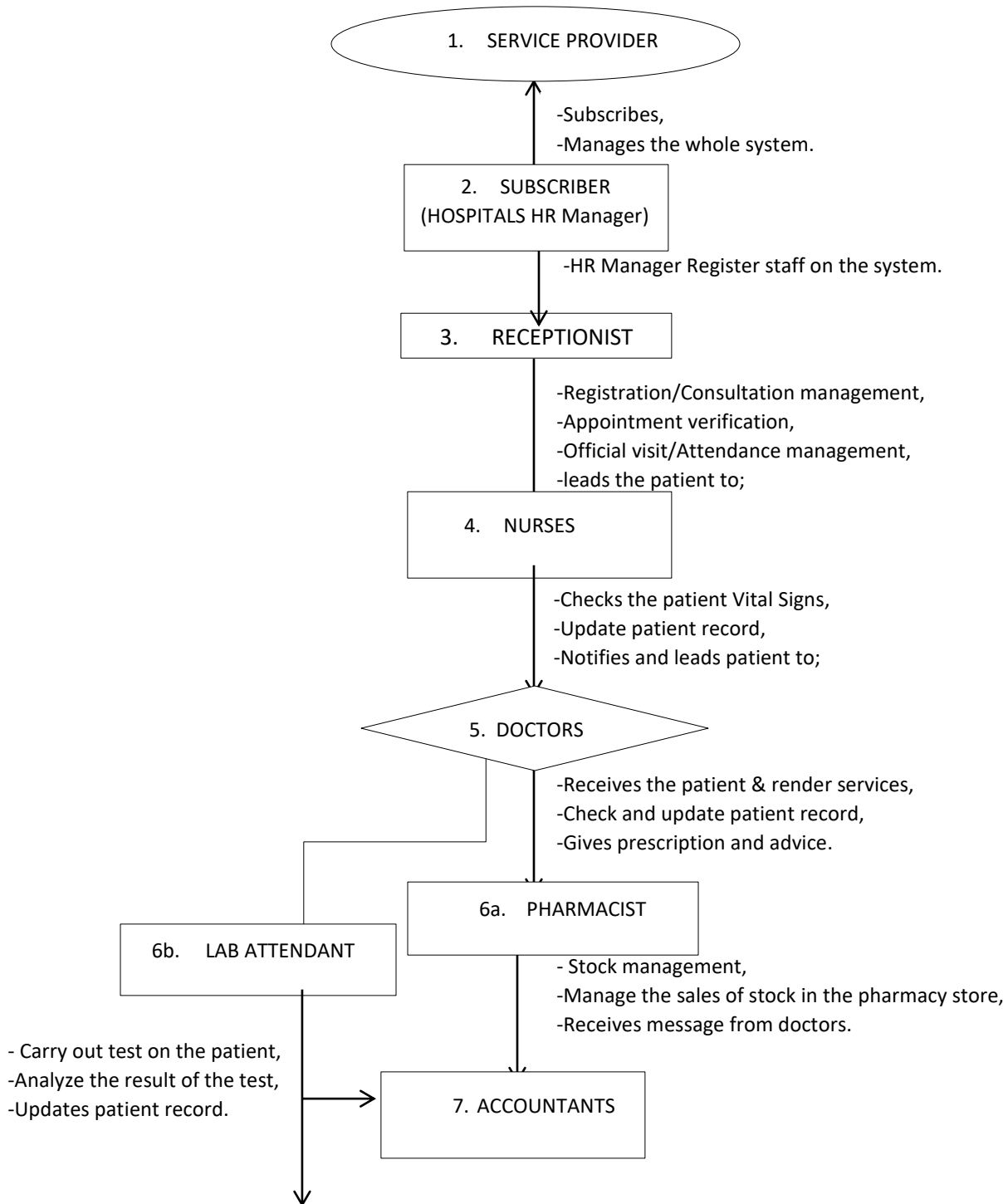
3.2 Methodology

The HDBMS was handled on both the front end and the back end; object oriented programming language such as C#, PHP, Java etc. can be used to accomplish the back end task while HTML, CSS and JavaScript can be used for the front end design. In this dissertation C#, HTML and CSS are used to handle both the front end and the backend.

In the C#, group of functions were coded in to a library called class library (generalClassLibrary), each of the functions in the class library are called in to the form tools. The whole project is classified into module according to the services to be rendered by the forms. The modules include registration module, login module, reference forms module, staff module, result module and stock and inventory module. Each module contained forms which serves similar but not the same function.

The registration module contains staff registration form and patient registration form, password reset form and subscriber form, also login module contains the login form, subscriber homepage, human resource page, the result module contains the vital signs form and the lab test form while the staff module contains doctors form, laboratory form, nurse form, patients form and lastly the stock inventory module contains the billing form, stock form and the sales form.

3.4 Architectural Plan (Normal System)



3.4 Continuation on Architectural Plan (Normal System)

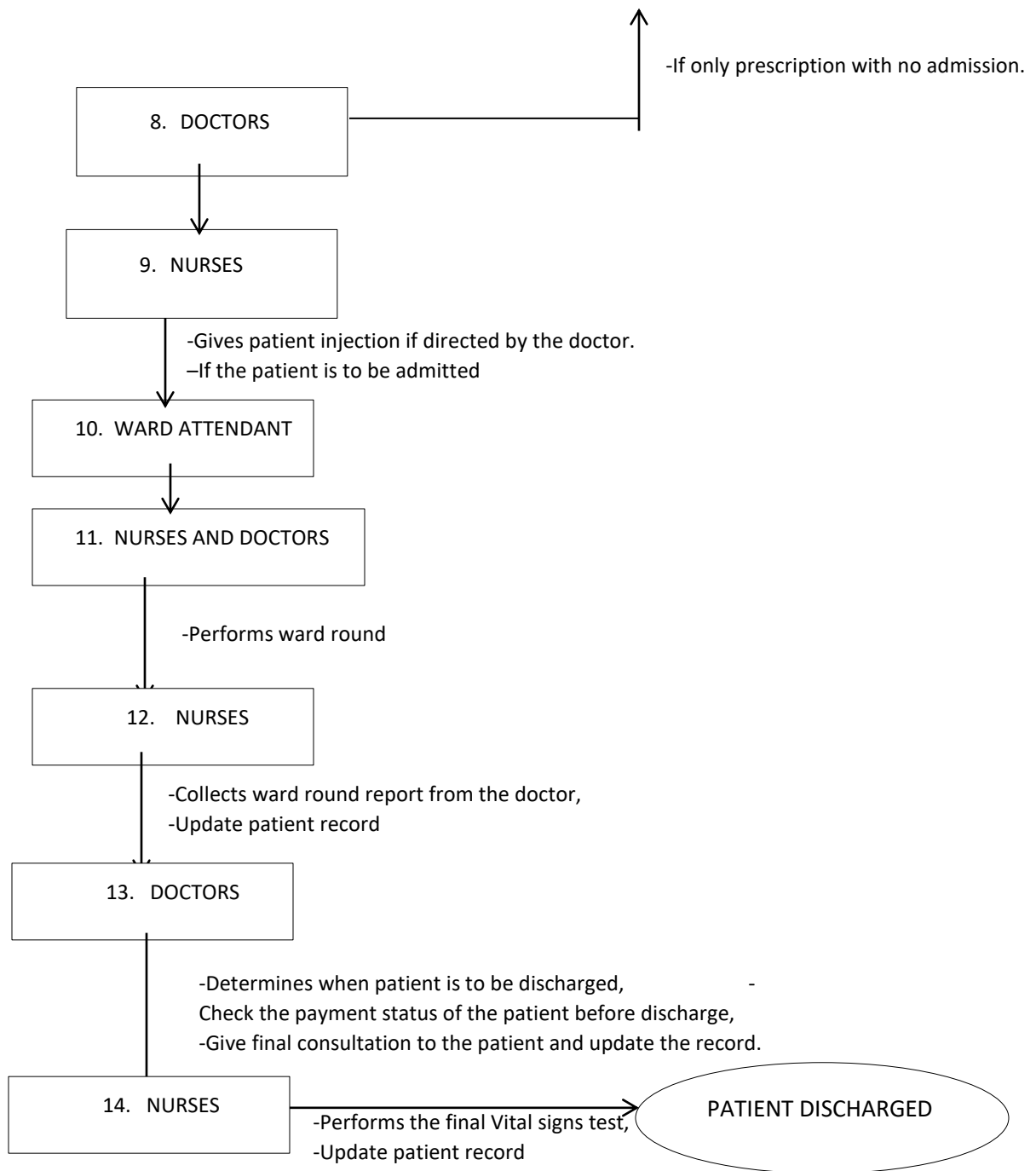


Fig 3.1 Architectural Plan (Normal System)

3.5 Architectural Plan (Emergency/Operation and Child delivery)

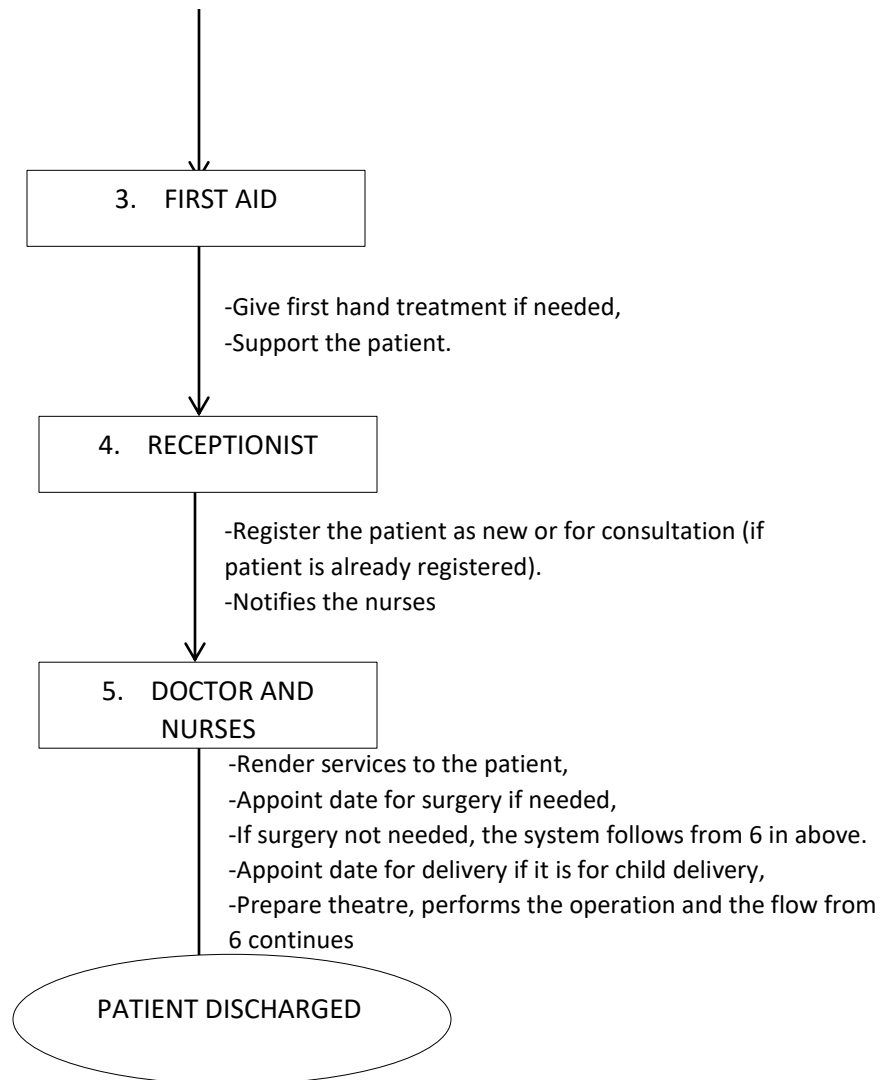


Fig 3.2 Architectural Plan (Emergency/Operation and Child delivery)

3.6 System Flowchart

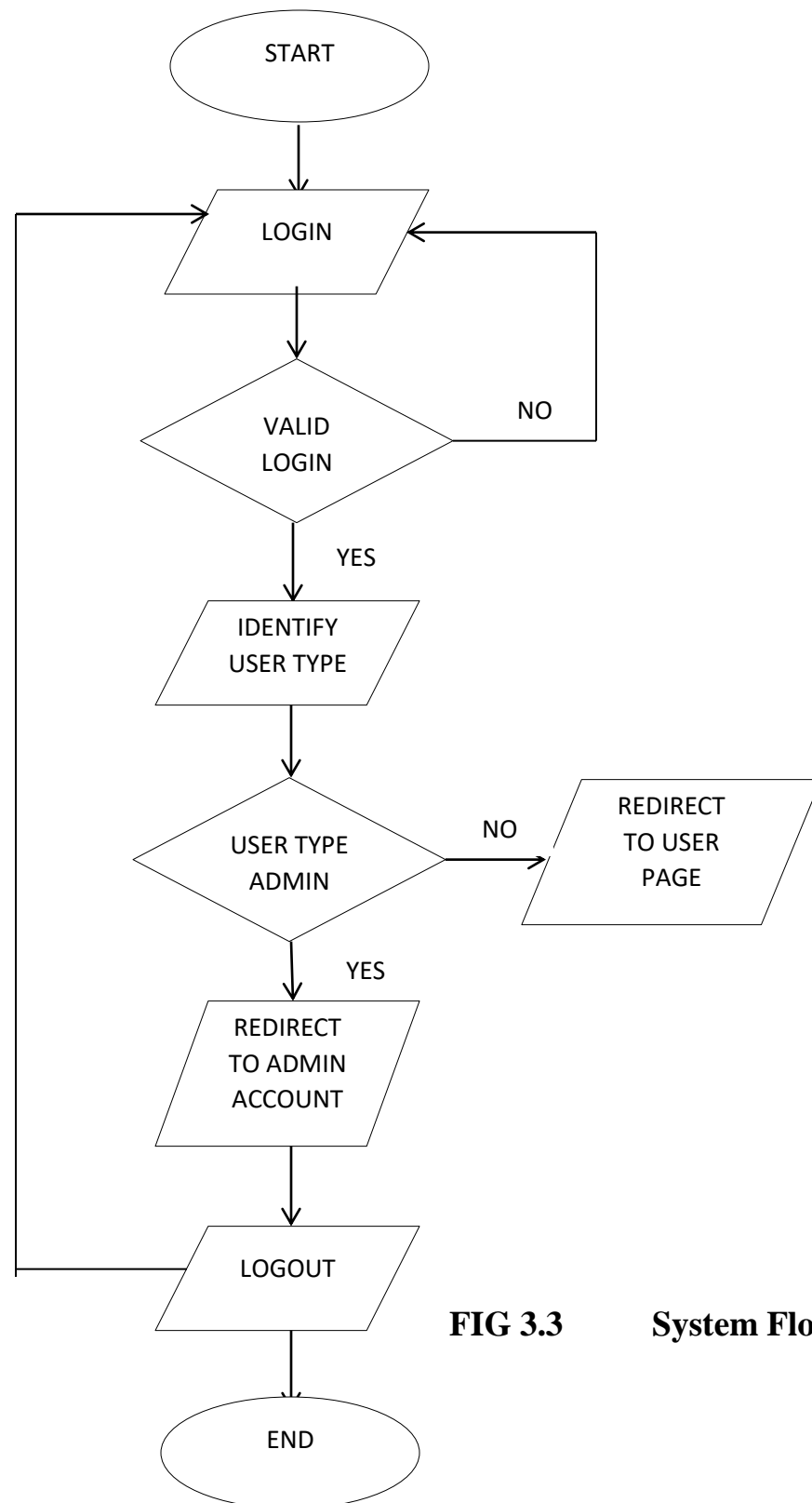


FIG 3.3 **System Flowchart**

3.7 Design Algorithm

The algorithm use in the design of the HDBMS is as follows;

Start

Registration

Staff registration to be done by admin

if staff registered

Staff login with his/her details to access the work page

if staff not registered

Staff meet with the admin for registration

if patient registered

Patient pay for consultation fees,

Pass the patient to the nurse,

Nurse Checks for Vital signs,

Nurse updates the patient record on the database from his/her page.

If patient not registered

Pay for registration,

Begin registration,

Process from *if patient registered* starts

Vital Signs Check

After vital signs check,

Nurse directs patient to the available doctor .

Doctor Consultation

Doctor checks the patient record using the patient ID search,

Doctor gives necessary advice and prescription,
Doctor update all details to the database from his/her work page,
Doctor make appointment with the patient,
Doctor determines if the patient is to be admitted or not.

If not admitted

Doctor notifies the lab and pharmaceutical store,
direct the patient to the necessary department.
Patient visits the account session for billing,
Financial manager issue receipts and update the patient record,
Patients go for paid investigations and/or drugs,

If admitted

Notifies the nurse,
Update the patient status,
Patient visits the account session for billing,
Financial manager issue receipts and update the patient record,
Takes the receipt to the nurse in charge,
Nurse allocates bed space,
Patient gets admitted.

Inpatients

Doctor render service,
Nurses treat the patient on doctor advice and needs of the patient,
Ward round is performed by doctors and nurses in charge,
Patient record of the ward round are updated by the nurses,
Doctor determines when discharge is due.

3.8 Input Design Specification

All the forms that require input were sanitized to take only values that are to be populated on the database; the input must be the same with the requirement on the database table. There are many types of input which varies from varchar, decimal, money, date and integer, which are commonly used as a requirement for taking input from the form. Each insert function into the database takes a new row on destination database table while the update function alters the data in the database table.

The functions mentioned above are coded in the class library called the generalClassLibrary, it is from this library the functions are called to use in the main program. For this class library function to work, stock procedure is designed to relate with the database table such that each public class function has its own stock procedure that makes it work. The stock procedure connects the class function and the database table parameters while the C# back end of the required form finally links the stock procedure to the form tools.

Furthermore, to make all the input work perfectly without throwing error from submission into the database try and catch function is implemented such that any required textbox or dropdown or other tools that takes input only take the input that is the same to specification on the database.

3.9 Hardware and Operating System Requirement

The devices needed to fully implement the HDBMS are as follows;

- 1) Laptop/Desktop PC
- 2) Display Unit
- 3) Laser Printer

- 4) Wi-Fi Router
- 5) Removable Disk

3.9.1 Hardware Specifications

Laptop/Desktop PC

- System Type; x32-bit / x64-bit
- ROM; 350GB Hard-Disk
- RAM; 4GB
- Processor; 1.2GHZ
- Language Version; English

Display Unit

- Resolution; 1366 x 768 (Recommended)
- Graphics; Intel 945 chipset or equivalent
- Graphic accelerator; nVidia or ATI

Laser Printer

- Print Resolution; 600 x 600 dpi
- Paper Type; A4

Wi-Fi Router

- Frequency; 2.4GHZ / 5GHZ
- Wireless Standard; 802.11
- Security; WPA /WPA 2/ WEP
- Antennas; one / two / three / four antennas

Removable Disk

- ROM; 100MB

3.9.2 Software Specifications

Laptop/Desktop PC

- Operating System; Windows 7/8/8.1/10
- Installation File; .exe
- .Net framework; 3.5
- Server Type; MySQL

CHAPTER 4

RESULT AND DISCUSSION

4.1 Subscriber Login Form

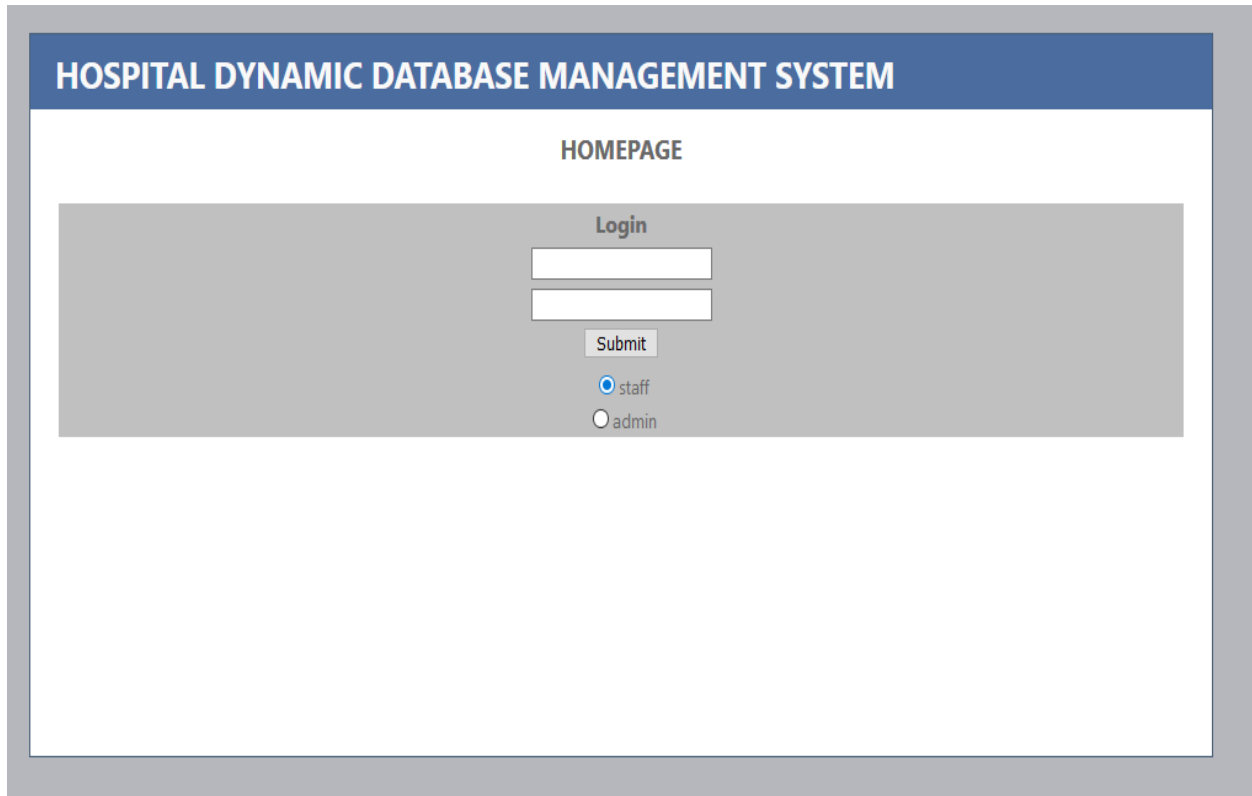
The screenshot displays a web application interface for the 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. The page is titled 'HOMEPAGE' and features a central 'Login' section. This section includes two text input fields for credentials, a 'Submit' button, and two radio button options: 'staff' and 'admin'. The 'admin' option is currently selected. The entire form is set against a light gray background within a white content area, which is framed by a blue header bar containing the system name.

Fig 4.1 **Subscriber Login Form (Source: Appendix 1, Login Form)**

The subscriber login form allows the hospital manager/human resource head/admin to access the entire HDBMS function by login in with the SN and the subscriber ID given by the service provider. The tool box that takes the login detail takes the value type that is the same with what is on the database such that the SN takes int type while the subscriber ID text box takes decimal. If the login is successful, the system redirect to the main login page. Thus the subscriber login page is the first page to be accessed by any hospital that is deploying the system for the first time

or renewing the subscription with the service provider, without this page passed, the user will not be able to access the next page (login page).

4.2 Login Form



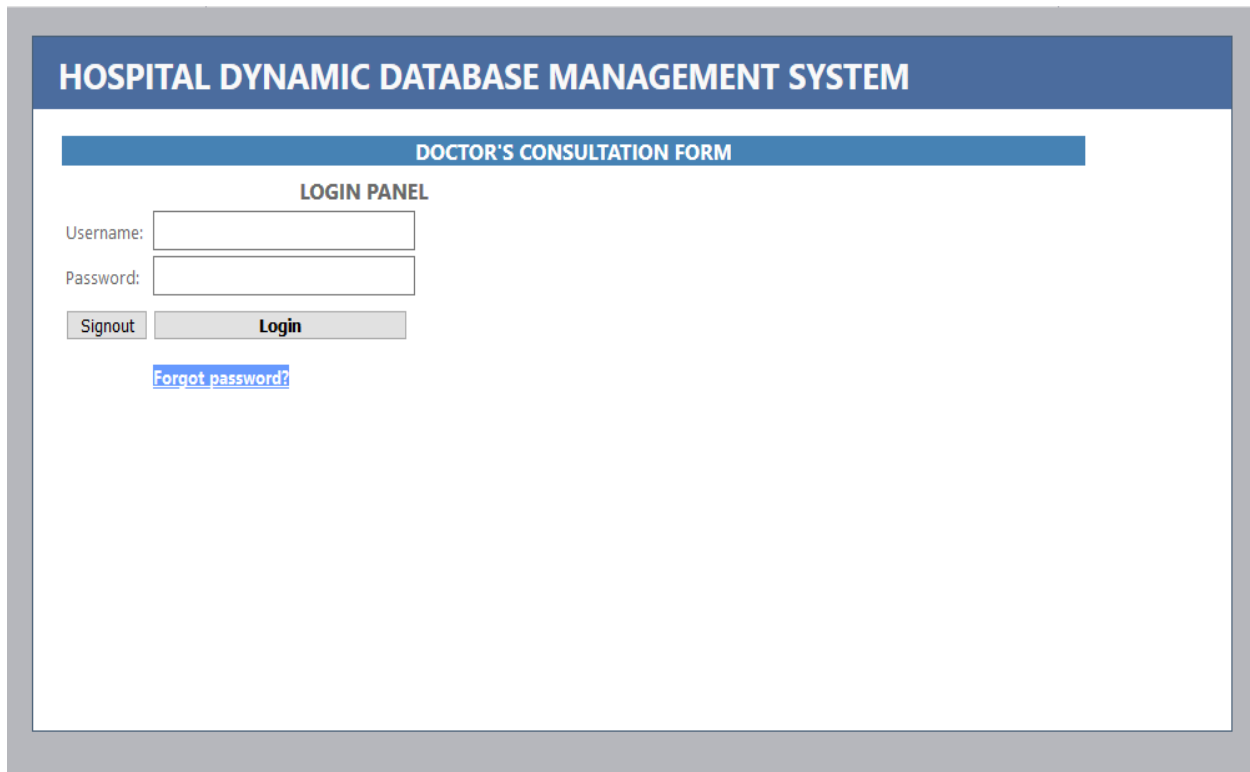
The screenshot displays a web application interface for a 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. At the top, a blue header bar contains the system name in white capital letters. Below this, the page is titled 'HOMEPAGE' in a smaller, bold, black font. The main content area features a light gray rectangular box with a 'Login' heading. Inside this box, there are two white input fields for text entry, followed by a 'Submit' button. Below the button, there are two radio button options: 'staff' (which is selected, indicated by a blue dot) and 'admin' (which is unselected, indicated by a white dot).

Fig 4.2 Login Form (Source: Appendix 1, Login Form)

The next page after the subscriber homepage is the login page, this page allows first the admin to login with the default password given by the service provider so as to register staff into the system for usage. The login button work in such a way that if pressed it checks the value in the staff ID box with the values on the staff database table, after matching the staff ID column on the table it checks if the password is correct with that row that has the staff ID, if so the system then picks

the staff ID redirect to the page best suitable for the staff to work such that it directs nurse to its page, doctor to consultation page etc.

4.3 Staff Login Panel



The screenshot displays a web interface for a 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. At the top, a blue header bar contains the system name in white capital letters. Below this, a white content area features a blue bar labeled 'DOCTOR'S CONSULTATION FORM'. Underneath, the text 'LOGIN PANEL' is centered. The login section includes two input fields: 'Username:' and 'Password:'. Below these fields are two buttons: 'Signout' and 'Login'. A blue hyperlink labeled 'Forgot password?' is positioned below the 'Login' button. The entire interface is enclosed in a grey border.

Fig 4.3 Staff Login Panel (Source: Appendix 1, Login Panel)

The login panel is the panel that first shows on any staff form, the panel request for the username and password. After the person enters his/her details, the program checks the staff registration table for the username and password, if this details are found to be true it picks the position of the owner of the username such that if the username is doctor the next page will display to welcome the person correct name and position.

4.4 Registration Form

The screenshot displays the 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM' interface. At the top, there are 'Home' and 'About' navigation links. Below this, the 'META HOSPITAL' logo is shown, followed by the 'STAFF REGISTRATION FORM' title. The form is divided into two main sections. The left section contains pre-filled information: 'Subscriber_ID: 32456' and 'Subscriber_Name: META HOSPITAL'. Below this are input fields for 'Staff ID:', 'Username:', 'Password:', 'Staff Name:', 'Gender:' (a dropdown menu with 'Select' as the current value), 'DOB:' (a date picker), 'Nationality:' (a dropdown menu with 'Afghanistan' as the current value), 'State of Origin:', 'Local Govt. of Origin:', 'Address:', 'Correspondent Address:', 'Mobile Number:', 'Department:' (a dropdown menu with 'Select' as the current value), and 'Email Address:'. A 'Submit Query' button is located to the right of the 'Username:' field. The right section features a large empty box for a profile picture, with the text 'Image quality: 99 x 93 Pixel' below it. Below the image box is a 'Select File:' label, a 'Browse...' button, and the text 'No file selected.'. Further down are input fields for 'Registration Date:', 'Next of Kin(NOK):', 'NOK Relationship:' (a dropdown menu with 'Select' as the current value), 'NOK Address:', 'NOK Mobile Number:', 'Highest Degree Obtained' (a dropdown menu with 'Select' as the current value), 'Position:' (a dropdown menu with 'Select' as the current value), and 'Role:' (a dropdown menu with 'Select' as the current value). At the bottom of the form, there are 'Clear' and 'Update' buttons on the left, and a 'Home' button on the right.

Fig 4.4.1 Staff Registration Form (Source: Appendix 1, Registration)

The staff registration form is designed such that only the admin have access to the page. On this page the admin can register the staff of the hospital with crucial details to be needed from the hospital about them, the data submitted in each hospital is been identified on the database table by the name and ID of the hospital such that once a staff uses the login details given by the admin it display the hospital where the worker is working and the ID of that hospital is shown in the

page as well. The registration page is to register all the staff in the hospital as all the departments and positions provided with many options on the registration form.

The HDBMS is designed to be dynamic such that all data are used on the forms or to perform a function are all taken from the database table, no hard coding on the backend of the system. On the staff registration from the dropdown boxes are bounded to the database tables to give likely options as shown below in fig;

The gender, nationality, state of origin, local government area, position, role, department, next of kin relationship and highest degree obtained are the dropdown fields that are bound from the database table to the staff registration form to make the form dynamic. The staff registration form submission is integrated to the table STAFF_REG_TBL and the stock procedure insertStaff_reg_tbl while the tables that feed the dropdown boxes are namely GENDER_TBL, DEPARTMENT_TBL, NOK_RELATIONSHIP_TBL, POSITION_TBL and ROLE_TBL.

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM

SUBSCRIBER FORM

Subscriber ID:

Subscriber Name:

Description:

Amount:

Paid:

Balance:

Subscription Date:

Expiry Date:

Subscriber Address:

Phone number:

Upload Logo:

Browse... No file selected.

Upload

Submit

Update




Fig 4.4.3 Subscriber Form (Source: Appendix 1, Registration)

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM

RONBEY

PATIENT REGISTRATION FORM

Subscriber_ID: 1034980

Subscriber_Name: RONBEY

Patient ID:

Patient Name:

Gender: Select

DOB:

Nationality: Afghanistan

State of Origin:

LGA:

Address:

Mobile Number:

Email Address:

Blood Group: Select

Genotype: Select

Label

Submit

Image quality: 99 x 93 Pixel

Select File: Browse... No file selected.

Next of Kin (NOK):

NOK Relationship: Select

NOK Address:

NOK Mobile Number:

Registration Date:

Staff Incharge: DAVID RUTH

Clear

Home




Fig 4.4.4 Patient Registration Form (Source: Appendix 1, Registration)

4.5 Consultation Form

The consultation form is one of the forms to be operation doctor of hospitals using the HDBMS, this form allows the doctor several functions that varies from checking past medical record of patients i.e. past consultations, vital signs, prescriptions, medical advice, last hospital used etc., giving patient prescription and treatment i.e. The system gives doctor possible treatment and prescription for the ailment enters into the system, giving medical advice and possible investigations to be carried out by patient. After these data are put into the input boxes the doctor can easily update the patient information by submitting it into the database table.

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM

META HOSPITAL

DOCTOR'S CONSULTATION FORM

Welcome Doctor **ADELADE AYOMI**

Akaun

32456

Patient Details

Patient Name: **Search**

Patient ID: [Collects Patient ID](#)

Gender: [Collect the Gender Here](#)

Recent Vital Signs Report: [Full Report Here](#)

Check Medical Record: **Check**

Fig 4.5.1 Consultation Form (Source: Appendix 1, Consultation Form)

<p>Consultations</p> <p>Diagnosis: <input type="text" value="Malaria"/> ▼</p> <p><input type="button" value="Add"/></p> <p>Symptoms: <input type="text"/></p> <p>Suggested Treatment: <input type="text"/></p> <p>List of Diagnosis: <input type="text"/></p> <p><input type="button" value="Remove"/></p>	<p>Investigations:</p> <p>Other Investigations: <input type="text" value="Radiography"/> ▼</p> <p>Department: <input type="text" value="X-RAY"/> ▼</p> <p>Investigations Type: <input type="text" value="Chest Scan"/> ▼</p> <p><input type="button" value="Add"/></p> <p>Investigation(s): <input type="text"/></p> <p><input type="button" value="Send"/></p> <p><input type="button" value="Remove"/></p>
<p>Prescriptions</p> <p>Prescription: <input type="text" value="Paracetamol"/> ▼</p> <p>Description: <input type="text"/></p> <p><input type="button" value="Add"/></p> <p>List of Prescription: <input type="text"/></p> <p><input type="button" value="Send"/></p> <p><input type="button" value="Remove"/></p>	<p>Remark: <input type="text"/></p> <p>Advice: <input type="text"/></p> <p>Date: <input type="text"/></p> <p><input type="button" value="Send"/></p> <p><input type="checkbox"/> Book Appointment</p> <p>Book Appointment: <input type="text"/></p> <p><input type="button" value="Send"/></p>

Fig 4.5.2 Consultation Form (Source: Appendix 1, Consultation Form)

The screenshot shows a 'Messaging' form with the following fields and controls:

- Sender:** ADELADE AYOMI
- Department ID:** A dropdown menu.
- Recipient ID:** A dropdown menu.
- Message:** A large text area for composing the message.
- Patient ID:** Collects the ID
- Send:** A button to send the message.
- Check General History?:** A link at the bottom of the form.

Fig 4.5.3 Consultation Form (Source: Appendix 1, Message Module)

Other feature on the consultation form include ability for doctor to login and sign out with his/her name, ability to search for patient by their ID on the database, ability to communicate with other departments as the form contains a message module that allows the doctor to send message to other staffs in the hospital and as well as other department.

Just as every other form, once the doctor login, if the login is successful the system picks the detail of the doctor from the database to display the doctor's name, ID, hospital name, picture and other details to be shown on the form. Thus, with consultation form doctor can make accurate prescription based on the past medical records and the present condition of any patient and also will be able to communicate with departments/staff in the hospital.

4.6 Nurse Form

Nurse form is designed to check the patient condition and put the details in the medical record. The form is only accessible by the nurses while it allows them to test for the vital signs (Body temperature, blood pressure, pulse rate, heart rate, respiration, oxygen saturation) and also record the ward round result. Patient must be on registered on the database before the form take any input or else it flags error as patient not registered on the platform, to make this possible patient ID searches are done by from the patient database table.

Fig 3.9.1 and Fig 3.9.2

The screenshot displays the 'NURSE'S SERVICE PAGE' of the 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. At the top, the system name is in a blue header. Below it, 'META HOSPITAL' is written in red. The page title 'NURSE'S SERVICE PAGE' is in a blue bar. A dashed blue box contains a welcome message: 'Welcome ADELADE AYOMI Akaun' with a small profile icon. Below this, a grey box titled 'Patient Details' contains a 'Patient ID' search field with a 'Search' button, a 'Patient Name' field, a 'Gender' field with a link 'Collect the Gender Here', and a 'Check Medical Record' button with a 'Check' button. A small profile icon is also present next to the search field.

Fig 4.6.1 Vital Signs (Source: Appendix 1, Vital Signs)

The image displays three web forms. The top left form, 'Vital Signs Upload', has a blue header and contains fields for Patient ID (with a 'Collects ID here' note), Upload (with a 'Browse...' button and 'No file selected.' text), Body Temperature, Blood Pressure, Pulse Rate, Heart Rate, Respirations, Oxygen Saturation, and Date (with a dropdown arrow). A 'Submit' button is at the bottom. The top right form, 'Ward Round Report Upload', also has a blue header and includes Patient ID (with a 'Collects ID here' note), Upload (with a 'Browse...' button and 'No file selected.' text), Date (with a dropdown arrow), Doctor on Duty, and a 'Submit' button. The bottom form, 'Messaging', has a grey header and includes fields for Sender (pre-filled with 'ADELADE AYOMI'), Department ID (dropdown), Recipient ID (dropdown), Message (text area), Patient ID (with a 'Collects the ID' note), and a 'Send' button. A link 'Check General History?' is at the bottom of the Messaging form.

Fig 4.6.2 Vital Signs (Source: Appendix 1, Vital Signs)

4.7 Laboratory Form

Laboratory form just like all other forms allows the lab attendants to upload result of investigation on patients into the database. The upper part of the form is the request from as sent from the doctor via the message module or the patient request based on doctor direction while the lower part is the field to upload the result from the investigation(s). The upload box is used as all the lab test will have a standard form of reporting the result and this can easily be uploaded in .doc file in any of the standard forms.

4.8 Patient Form

The patient form is a unique form generated for individual patients, the form contains the basic details about the patient coupled with medical history and activities of patients. The form can be accessed by only the doctor and the patient ID is used to search for patient record, the form contains all the lab report history, vital sign reports, past doctor's report, prescription report, services report and billing report.

With the patient form that can be accessed by doctors in all the subscribed hospital it is easy to check the past and present condition of a patient and use it to make prescription and give advice to the patient. Also the billing information shows the payment status of the patient and the transaction history for services and products used in the hospital.

The screenshot displays the 'PATIENT'S FORM' interface within the 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. The system's title bar includes a 'Log In' link. Navigation tabs for 'Home' and 'About' are visible. The form is titled 'PATIENT'S FORM' and features a 'Welcome' message, a 'Patient ID' input field, and a 'Patient Details' section with fields for Gender, Phone Number, Email Address, Blood Group/Genotype, Status, and Payment Status. Below this is a 'Reports' section containing two report grids: 'Lab Report Grid' and 'Vital Signs Report Grid', both of which currently show 'No Data To Display'.

Fig 4.8.1 Patient form (Source: Appendix 1, Patient Form)

The image shows a patient form layout with four report grids stacked vertically. Each grid is a light blue rectangle with the text "No Data To Display" in the top-left corner. The grids are labeled as follows:

- Doctors Report Grid
- Prescription Report Grid
- Services Report Grid
- Billing Report Grid

The form is framed by a grey border on the left and right sides.

Fig 4.8.2 Patient form (Source: Appendix 1, Patient Form)

4.9 Billing Form and Sales Form

The billing form is handled by the financial/accountant manager to record the billing information of services to be taken by patient. This billing form populates the grid on patient profile form as soon as submission button is done and also update the payment status of the patient.

The screenshot displays a web application interface for a hospital database management system. At the top, a blue header bar contains the text "HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM". Below this, a red banner reads "META HOSPITAL". A prominent blue bar with the text "BILLING FORM" is positioned below the red banner. The main content area is enclosed in a dashed blue border. It features a "Staff Incharge:" label followed by the text "Akaun" and a small icon of a person. Below this, there is a large empty rectangular box. Further down, a section titled "Patient Details:" is shown. It includes a "Patient ID:" label, a text input field, and a "Search" button. To the right of the input field is a small icon of a document. Below the "Patient ID:" field, there are labels for "Patient Name:", "Phone Number:", "Address:", "Gender:", and "Status:". Each of these labels is followed by the word "Label" in blue text. The "Status:" label is followed by the text "Collect Status from Database" in blue. To the right of these labels is a large empty rectangular box.

Fig 4.9.1 Billing Form (Source: Appendix 1, Billing Form)

Bills		Other Services:	
Bills:	<input type="checkbox"/> Card <input type="checkbox"/> Consultation	Other Services:	<input type="text"/>
Description:	<input type="text"/>	Price:	<input type="text"/>
Price:	<input type="text"/>	Trans Date:	<input type="text"/>
Quantity:	<input type="text"/>	Trans Time:	<input type="text"/>
Amount:	<input type="text"/>	<input type="button" value="Add"/>	
Grand Total:	<input type="text"/>	List to be paid for:	<input type="text"/>
<input type="button" value="Add"/>			

Insurance / Discount(%):	<input type="text"/>
Grand Total:	<input type="text"/>
Amount Paid:	<input type="text"/>
Balance:	<input type="text"/>
<input type="button" value="Make Payment"/>	

Fig 4.9.2 Billing Form (Source: Appendix 1, Billing Form)

The sales form is more like the billing form only that this is mostly used for the pharmaceutical session to record the stock sales from the store, once pharmacist receives prescription from the doctor/patient, he fills the sales form as soon as the patient makes payment for the items bought from the store. This detail is updated into the billing grid of the patient as well.

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM [Log In]

Home About

META HOSPITAL

SALES FORM

Sales ID:

Stock:

Item Price:

Quantity:

Add Item

Description:

Item List:

Remove

Grand Total:

Patient Name | ID:

Staff Name | ID: Akaun

Trans Date:

Trans Time:

Complete Sales

Cancel

Fig 4.9.3 Sales Form (Source: Appendix 1, Sales Form)

4.10 Stock Form

Stock form is designed to account for all the entire item inside the store and as well keep track of sales and order made by the store. The stock from is supported with the stock reference form, the category reference form is used in classifying the items in the store and it is works on the principle of simple arithmetic of addition and subtraction such that as sales is made the quantity of item sold is subtracted from the total number of that item in the store.

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM [Log In]

Home About

META HOSPITAL

STOCK FORM

Stock Details

Category:

Sub Category:

Sub Sub Category:

Stock ID:

Stock Name:

Description:

Quantity:

Unit Price:

Selling Price:

Proposed Profit:

Staff Incharge: Akaun

Registration Date:

Expiry Date:

Fig 4.10 Stock Form (Source: Appendix 1, Stock Form)

4.11 Reference Form

Reference forms are forms used to update information on the database, these forms are very essential in adding new data into the database and it allows more dynamic input of data. The reference form designed in this application are bed reference form, category reference form, country reference form, department reference form, price list reference form, diseases and symptoms reference form.

The reference form also aid suggestion when bounded to text or dropdown tool box on form, the disease and symptoms reference table from the database is used on the consultation page which helps doctors with making good prescription for any ailment chosen that is on the database. Other reference forms serves major purpose in the HDBMS application such as category reference form is used on the stock

inventory, bed reference form for the inpatients, price list for both stock inventory and billing form, country reference form for registration pages and department reference form on registration pages and messaging module.

The screenshot shows the 'META HOSPITAL' interface within the 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. The page has a blue header with 'Home' and 'About' links, and a '[Log In]' button. Below the header, a blue bar displays 'META HOSPITAL' in red and 'BED REFERENCE FORM' in white. The main content area contains a form with the following fields: 'Section:' (a dropdown menu), 'Ward:' (a text input), 'Room:' (a text input), and 'Bed:' (a text input). An 'Add' button is positioned below the 'Bed' field.

Fig 4.11.1 Reference Form (Source: Appendix 1, Reference Form)

The screenshot shows the 'META HOSPITAL' interface within the 'HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM'. The page has a blue header with 'Home' and 'About' links, and a '[Log In]' button. Below the header, a blue bar displays 'META HOSPITAL' in red and 'Category Form' in white. The main content area contains three sections, each with a blue header bar: 1. 'Category' section: 'Category Name:' (text input), 'Clear' button, 'Add Category' button, and 'Next' button. 2. 'Sub Category' section: 'Category Name:' (dropdown), 'Sub Category Name:' (text input), 'Clear' button, 'Add Sub Category' button, 'Next' button, and 'Back' button. 3. 'Sub Sub Category' section: 'Category Name:' (dropdown), 'Sub Category Name:' (dropdown), 'Sub Sub Category Name:' (text input), 'Clear' button, 'Add Sub Sub Cat' button, 'Next' button, and 'Back' button.

Fig 4.11.2 Reference Form (Source: Appendix 1, Reference Form)

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM [[Log In](#)]

Home About

META HOSPITAL

DISEASE REFERENCE FORM

Disease:

Microbes:

Causes:

Symptoms:

Treatment:

Fig 4.11.3 Reference Form (Source: Appendix 1, Reference Form)

HOSPITAL DYNAMIC DATABASE MANAGEMENT SYSTEM [[Log In](#)]

Home About

META HOSPITAL

PRICE LIST

Department:

Services:

Price:

Fig 4.11.4 Reference Form (Source: Appendix 1, Reference Form)

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Hospital database management system can be made fully dynamic without any form of hardcoding at the backend part of the code; this can be effectively realized by making use of relational database tables, class libraries and database procedures. The dynamic HDBMS serves so many functions that vary from sensitive staff page such as on the consultation page, statistical analysis of the database data, also messaging module should be implemented in hospital database management system for easy communication among the staff as this will increase the service productivity of the health care sector.

5.2 Recommendation

To enhance the feature of the HDBMS designed, the financial aspect of it needed to be worked on so as to make the management system fully dynamic. And as well API integrations such as thumb printing and payment gateway can be added to make the system more effective.

The language used also gives a friendly extension to various platforms like TOR and Blockchain Technology if planned to be incorporated in the future.

5.3 Contribution to Knowledge.

Some of the hospitals in the country operate on their respective HDBMS, but with this HDBMS software which creates a centralized platform for all the hospital to register and easily monitor the individual hospital record. Also the messaging module that allows active communication between hospital staff and analysis page that gives basic statistics of the hospital data.

REFERENCES

OlamideO, Olusanya, Elegbede W Adedayo, and Ogunseye a Abiodun. 2015. “Design and Implementation of Hospital Management System Using Java.” *IOSR Journal of Mobile Computing & Application* 2(1): 2394–42. www.iosrjournals.org.

Balaraman, Premkumar, and Kalpana Kosalram. 2013. “E –Hospital Management & Hospital Information Systems – Changing Trends.” *International Journal of Information Engineering and Electronic Business* 5(1): 50–58. <http://www.mecs-press.org/ijieeb/ijieeb-v5-n1/v5n1-6.html>.

Ngorsed, Manoon, and Poonphon Suesaowaluk. “HOSPITAL SERVICE QUEUE MANAGEMENT SYSTEM 1 Introduction 2 Literature Review.” : 550–59.

“International Journal of Database Management Systems (IJDMS).” 2011. 5705. <http://airccse.org/journal/ijdms/ijdms.html>.

Asabe, S.a, and N.D Oye. 2013. “Hospital Patient Database Management System.” *International Journal of Advanced Computer Technology* 2(3): 65–72.

Ilo, S F, Igbajar Abraham, and Acholonu Joyce C. 2015. “Designing A Web Based Hospital Management System For MOUAU Clinic.” *International Journal of Trend in Research and Development* 2(6): 286–93. www.ijtrd.com.

Academy, Huawei I C T. “Huawei ICT Competition Student Registration Guide V2 . 0.” : 0–6.

Cui Xiu-li, Li Feng-yun, Wang Xin, and Li Peng. 2008. “Design and Implementation of Intelligent Power Management System.” *2008 Chinese Control and Decision Conference* 5(1): 329–34. <http://ieeexplore.ieee.org/document/4597324/>.

Tripathee, Deepak. 2016. “Hospital Database Management System.” *International Journal of Computer Science and Mobile Computing* 54(4): 71–73. <http://www.ijcsmc.com/docs/papers/April2016/V5I4201602.pdf>.

Gadhari, Digvijay H, and Yadnyesh P Kadam. 2016. “Hospital Management System 1 1,2,3.” (11).

Vsai. 2009. “Hospital Management System.” *Vasudha SoftAim Innovations Pvt. Ltd.* http://www.softaiminnovations.com/web/index.php?option=com_content&view=article&id=19&Itemid=28.

Science, Computer, and Management Studies. 2015. “Comparison of HDBMS , NDBMS , RDBMS and OODBMS.” 7782: 119–27.

Rana, Vicky. "CS 561 Project 4 – Hospital Vicky Rana." : 1–37.

Mahmood, Nadeem, Aqil Burney, Zain Abbas, and Kashif Rizwan. 2012. "Data and Knowledge Management in Designing Healthcare Information Systems." *International Journal of Computer Applications* 50(2): 34–39. <http://research.ijcaonline.org/volume50/number2/pxc3880798.pdf>.

Patel, Sanskruti, and Atul Patel. 2016. "A Big Data Revolution in Health Care Sector: Opportunities, Challenges and Technological Advancements." *International Journal of Information Sciences and Techniques (IJIST)* 6(1): 155–62. <http://aircconline.com/ijist/V6N2/6216ijist16.pdf>.

Gacheri, Nkanata Mercy. 2015. "Use of Hospital Management Information Systems among Healthcare Workers at Kenyatta National and Mater Hospitals."

Oyedeki, Stephen, Augustine Onwuegbukiwe Isichei, and Adekunle Ogunfidodo. 2014. "Performance of Some Local Nigerian Turfgrasses in Sole and Mixed Stands." *Turkish Journal of Field Crops* 19(1): 101–7.

Campbell, Robert J. 2004. "Database Design: What HIM Professionals Need to Know." *Perspectives in health information management / AHIMA, American Health Information Management Association* 1: 6.

Tripathi, Deepak. 2016. "Hospital Database Management System." *International Journal of Computer Science and Mobile Computing* 54(4): 71–73. <http://www.ijcsmc.com/docs/papers/April2016/V5I4201602.pdf>.

"Learn Object Oriented Programming (OOP) in PHP."

Datathe, Understanding, World Health, and Statistics Series. 2018. "Full-Text." : 3–5.

APPENDIX I

BACKEND CODES

Login forms

```
public partial class Login_Module_Homepage : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        if (radAdmin.Checked == true)
        {
            hdbmsClass lib = new hdbmsClass();
            DataSet ds = new DataSet();
            string msg = "";
            int sn = Convert.ToInt32(txtSN.Text);
            decimal subscriberID = Convert.ToDecimal(txtSubscriberID.Text);

            ds = lib.fetchSubscriberBySN_subID(sn, subscriberID, ref msg);

            if (msg != "" || ds.Tables.Count == 0)
            {
                lblErrorMsg.Text = "unable to fetch subscriber as a result of " + msg;
                MessageBox.Show("Incorrect Details!!!");
                return;
            }
            Response.Redirect("~/Login_Module/Human_Resource_Page.aspx");
        }
        else if (radStaff.Checked == true)
        {
            hdbmsClass lib = new hdbmsClass();
            DataSet ds = new DataSet();
            string msg = "";
            string userName = txtSN.Text;
            string password = txtSubscriberID.Text;

            ds = lib.fetchStaffByUsername_Password(userName, password, ref msg);

            if (msg != "" || ds.Tables.Count == 0)
            {
                lblErrorMsg.Text = "unable to fetch staff as a result of " + msg;
                MessageBox.Show("Incorrect Details!!!");
                return;
            }
        }
        try
        {
            string a = txtSN.Text.Substring(0, 3);
            if (a == "DOC")
            {
                Response.Redirect("~/Staff_Module/Doctors_Form.aspx");
            }
            else if (a == "NUR")
            {
                Response.Redirect("~/Staff_Module/Nurse_Form.aspx");
            }
            else if (a == "LAB")
            {
                Response.Redirect("~/Staff_Module/Laboratory_Form.aspx");
            }
            else if (a == "ACB")
            {
                Response.Redirect("~/Stock_Inventory_Module/Billing_Form.aspx");
            }
            else if (a == "HUM")
            {
                Response.Redirect("~/Login_Module/Human_Resource_Page.aspx");
            }
        }
    }
}
```

```

        else if (a == "RCP")
        {
            Response.Redirect("~/Registration_Module/Patient_Reg_Form.aspx");
        }
        else
        {
            MessageBox.Show("You are not allowed on this system!!!");
        }
    }
    catch
    {
        MessageBox.Show("Wrong login detail, try again!");
    }

    Response.Redirect("~/Staff_Module/Doctors_Form.aspx");
}
}
}

```

Login Panel

```

protected void btnLogin_Click(object sender, EventArgs e)
{
    try
    {
        hdbmsClass lib = new hdbmsClass();
        DataSet ds = new DataSet();
        string msg = "";
        string userName = txtUsername.Text;
        string password = txtPassword.Text;

        ds = lib.fetchStaffByUsername_Password(userName, password, ref msg);

        if (msg != "" || ds.Tables.Count == 0)
        {
            lblErrorMsg.Text = "unable to fetch staff as a result of " + msg;
            MessageBox.Show("Incorrect Details!!!");
            return;
        }
        string a = txtUsername.Text.Substring(0, 3);
        if (a == "DOC")
        {
            Panel10.Visible = true;
            Panel5.Visible = false;
            Panel11.Visible = true;
            Panel111.Visible = true;

            lblStaffname.Text = ds.Tables[0].Rows[0]["STAFF_NAME"].ToString();
            lblStaffID.Text = ds.Tables[0].Rows[0]["STAFF_ID"].ToString();
            imgPassport.ImageUrl = ds.Tables[0].Rows[0]["IMAGE_UPLOAD"].ToString();
            lblSubscriberName.Text = ds.Tables[0].Rows[0]["SUBSCRIBER_NAME"].ToString();
            lblSubscriberID.Text = ds.Tables[0].Rows[0]["SUBSCRIBER_ID"].ToString();
            lblStaffname2.Text = ds.Tables[0].Rows[0]["STAFF_NAME"].ToString();
        }
        else
        {
            MessageBox.Show("Not allowed on this system!!!");
        }
    }
    catch
    {
        MessageBox.Show("Wrong login detail, try again!");
        Response.Redirect("~/Staff_Module/Doctors_Form.aspx");
    }
}

```

Registration

```

public partial class Registration_Module_Staff_Reg_Form : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            dropNationality.DataSource = country_name_list();
            dropNationality.DataBind();
        }
    }
    public static List<string> country_name_list()
    {
        List<string> Culturelist = new List<string>();
        CultureInfo[] getcultureinfo = CultureInfo.GetCultures(CultureTypes.SpecificCultures);
        foreach(CultureInfo getculture in getcultureinfo)
        {
            RegionInfo getreginfo = new RegionInfo(getculture.LCID);
            if (!(Culturelist.Contains(getreginfo.EnglishName)))
            {
                Culturelist.Add(getreginfo.EnglishName);
            }
        }
        Culturelist.Sort();
        return Culturelist;
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        if (txtStaffName.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the staff name");
            return;
        }
        else if (txtAddress.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the address");
            return;
        }
        else if (txtUsername.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the username");
            return;
        }
        else if (txtPassword.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the password");
            return;
        }
        else if (txtEmailAddress.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the email");
            return;
        }
        else if (txtMobileNumber.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the mobile number");
            return;
        }
        else if (txtNOK.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the next of kin name");
            return;
        }
        else if (txtNOKMobile.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the next of kin number");
            return;
        }
        FileUpload1.SaveAs(Server.MapPath("Images").ToString() + @"\" + FileUpload1.FileName);
        PassportImg.ImageUrl = @"~\Images\" + FileUpload1.FileName;
        PassportImg.DataBind();
    }
}

```



```

hdbmsClass lib = new hdbmsClass();
hdbmsClass lib2 = new hdbmsClass();
string msg = "";
decimal subscriberID = Convert.ToDecimal(lblSubscriberID.Text);
string subscriberName = lblSubscriberName.Text;
string staffID = txtStaffID.Text;
string staffName = txtStaffName.Text;
string gender = dropGender.Text;
DateTime dob = Convert.ToDateTime(dateDOB.Value);
string nationality = dropNationality.Text;
string state = dropState.Text;
string lg = dropLG.Text;
string address = txtAddress.Text;
string correspondentAddress = txtCorrespondent.Text;
string mobileNumber = txtMobileNumber.Text;
string emailAddress = txtEmailAddress.Text;
string department = dropDepartment.Text;
string positionLevel = dropPosition.Text;
DateTime registrationDate = Convert.ToDateTime(dateRegistration.Value);
string nextOfKin = txtNOK.Text;
string nokRelationship = dropNOKRel.Text;
string nokAddress = txtNOKAddress.Text;
string nokMobileNumber = txtNOKMobile.Text;
string highestDegreeObtained = dropDegree.Text;
string role = dropRole.Text;
string imageUrl = PassportImg.ImageUrl;
string userName = txtUsername.Text;
string password = txtPassword.Text;
lib.insertStaff_reg_tbl(subscriberID, subscriberName, staffID, staffName, gender, dob, nationality, state, lg, address, correspondentAddress,
mobileNumber, emailAddress, department, positionLevel, registrationDate, nextOfKin, nokRelationship, nokAddress, nokMobileNumber,
highestDegreeObtained, role, imageUrl, userName, password, ref msg);
lib2.insertMail_reg_tbl(subscriberID, subscriberName, staffID, staffName, gender, nationality, department, mobileNumber, emailAddress,
userName, password, ref msg);
if (msg != "")
{
    lblMessage.Text = "unable to register subscriber as a result of " + msg;
    MessageBox.Show("Registration failed!!!");
    return;
}
else
{
    MessageBox.Show("Successful!");
    Response.Redirect("~/Registration_Module/Staff_Reg_Form.aspx");
}
}

```

Consultation Form

```

public partial class Staff_Module_Doctors_Form : System.Web.UI.Page
{
}
protected void Button12_Click(object sender, EventArgs e)
{
    if (dropDiagnosis.SelectedIndex > -1)
    {
        diagnosisList.Items.Add(dropDiagnosis.SelectedItem);
        diagnosisList.ClearSelection();
    }
}
protected void Button19_Click(object sender, EventArgs e)
{
    if (diagnosisList.SelectedIndex > -1)
    {
        diagnosisList.Items.RemoveAt(diagnosisList.SelectedIndex);
        diagnosisList.ClearSelection();
    }
}

```

```

}
protected void DropDownList4_SelectedIndexChanged(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    DataSet ds = new DataSet();
    string msg = "";

    string investigationCategory = departmentDrop0.SelectedItem.ToString();
    string investigationType = investigationTypeDrop.SelectedItem.ToString();
    ds = lib.fetchInvestigationByInvestigationType(investigationCategory, investigationType, ref msg);

    if (msg != "")
    {
        lbl.Text = "unable to fetch investigation as a result of " + msg;
        MessageBox.Show("Error fetching investigation!!!");
        return;
    }
    investigationDrop.DataSource = ds;
    investigationDrop.DataTextField = "INVESTIGATION";
    investigationDrop.DataValueField = "INVESTIGATION";
    investigationDrop.DataBind();
    investigationDrop.Items.Insert(0, "Select Investigation");
    investigationDrop.Items[0].Value = "Select";
}
protected void Button20_Click(object sender, EventArgs e)
{
    if (investigationDrop.SelectedIndex > -1)
    {
        investigationsList.Items.Add(investigationDrop.SelectedItem);
        investigationsList.ClearSelection();
    }
}
protected void Button21_Click(object sender, EventArgs e)
{
    if (investigationsList.SelectedIndex > -1)
    {
        investigationsList.Items.RemoveAt(investigationsList.SelectedIndex);
        investigationsList.ClearSelection();
    }
}
protected void Button24_Click(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    //hdbmsClass lib1 = new hdbmsClass();
    string msg = "";
    //decimal subscriberID = lib.GenerateTransID(ref msg);

    string patientID = txtPatientID.Text;
    DateTime consultationDate = Convert.ToDateTime(dateConsultaion.Value);
    string diagnosis = diagnosisList.Text;
    string investigations = investigationsList.Text;
    string prescriptions = prescriptionsList.Text;
    string remark = txtRemark.Text;
    string advice = txtAdvice.Text;
    string doctorID = lblStaffID.Text;

    lib.insertDoctor_report_tbl(patientID, consultationDate, diagnosis, investigations, prescriptions, remark, advice, doctorID, ref msg);

    if (msg != "")
    {
        lblError.Text = "unable to update doctors report as a result of " + msg;
        MessageBox.Show("Registration failed!!!");
        return;
    }
    MessageBox.Show("Successful!");
}
protected void dropDiagnosis_SelectedIndexChanged(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    DataSet ds = new DataSet();

```

```

string msg = "";

string diagnosis = dropDiagnosis.SelectedItem.ToString();
string symptoms = symptomsTxt.Text;
string treatment = txtTreatment.Text;

ds = lib.fetchSymptomsByDisease(diagnosis, ref msg);

if (msg != "")
{
    lblError.Text = "unable to fetch treatment as a result of " + msg;
    MessageBox.Show("Error fetching treatment!!!");
    return;
}

symptomsTxt.Text = ds.Tables[0].Rows[0]["SYMPTOMS"].ToString();
txtTreatment.Text = ds.Tables[0].Rows[0]["TREATMENT"].ToString();
}

protected void Button3_Click(object sender, EventArgs e)
{
    if (txtPatientID.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Enter the Patient ID");
        return;
    }
    try
    {
        hdbmsClass lib = new hdbmsClass();
        DataSet ds = new DataSet();
        string msg = "";
        string patientID = txtPatientID.Text;
        ds = lib.fetchPatientByID(patientID, ref msg);
        if (msg != "" || ds.Tables.Count == 0)
        {
            lblErrormsg.Text = "unable to fetch patient as a result of " + msg;
            MessageBox.Show("Incorrect Details!!!");
            return;
        }
        lblPatientName.Text = ds.Tables[0].Rows[0]["PATIENT_NAME"].ToString();
        txtGender.Text = ds.Tables[0].Rows[0]["GENDER"].ToString();
        imgPassport.ImageUrl = ds.Tables[0].Rows[0]["IMAGE_UPLOAD"].ToString();
    }
    catch
    {
        MessageBox.Show("Incorrect Details!!!");
    }
}

protected void departmentDrop0_SelectedIndexChanged(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    DataSet ds = new DataSet();
    string msg = "";

    string investigationCategory = departmentDrop0.SelectedItem.ToString();
    ds = lib.fetchInvestigationTypeByInvestigationCategory(investigationCategory, ref msg);

    if (msg != "")
    {
        lbl.Text = "unable to fetch investigation type as a result of " + msg;
        MessageBox.Show("Error fetching investigation type!!!");
        return;
    }

    investigationTypeDrop.DataSource = ds;
    investigationTypeDrop.DataTextField = "INVESTIGATION_TYPE";
    investigationTypeDrop.DataValueField = "SN";
    investigationTypeDrop.DataBind();
}

```

```

        investigationTypeDrop.Items.Insert(0, "Select Investigation Type Name");
        investigationTypeDrop.Items[0].Value = "Select";
    }
protected void Button4_Click(object sender, EventArgs e)
{
    if (txtPatientID.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Enter the Patient name");
        return;
    }

    try
    {
        Response.Redirect("~/Staff_Module/Patients_Form.aspx");
    }

    catch
    {
        MessageBox.Show("Incorrect Details!!!");
    }
}
}

```

Vital Signs

```

public partial class Staff_Module_Nurse_Form : System.Web.UI.Page
{
    protected void Button17_Click(object sender, EventArgs e)
    {
        FileUpload1.SaveAs(Server.MapPath("results").ToString() + @"\" + FileUpload1.FileName);

        hdbmsClass lib = new hdbmsClass();
        string msg = "";
        string patientID = lblPatientID.Text;
        string patientName = lblPatientName.Text;
        DateTime testDate = Convert.ToDateTime(dateTest.Value);
        string bodyTemperature = txtBodyTemperature.Text;
        string bodyPressure = txtBodyPressure.Text;
        string pulseRate = txtPulseRate.Text;
        string heartRate = txtHeartRate.Text;
        string respiration = txtRespiration.Text;
        string oxygenSaturation = txtOxygenSaturation.Text;
        string staffID = lblStaffID.Text;
        string hospitalName = lblSubscriberName.Text;
        string result = FileUpload1.FileName;

        lib.insertVital_signs_tbl(patientID, patientName, result, testDate, bodyTemperature, bodyPressure, pulseRate, heartRate, respiration,
        oxygenSaturation, staffID, hospitalName, ref msg);

        if (msg != "")
        {
            lblMessage.Text = "unable to update vital signs report as a result of " + msg;
            MessageBox.Show("Registration failed!!!");
            return;
        }
        MessageBox.Show("Successful!");
    }
protected void Button18_Click(object sender, EventArgs e)
{
    FileUpload2.SaveAs(Server.MapPath("results").ToString() + @"\" + FileUpload2.FileName);

    hdbmsClass lib = new hdbmsClass();
    string msg = "";
    //decimal subscriberID = lib.GenerateTransID(ref msg);

    string patientID = lblPatientID.Text;

```

```

string patientName = lblPatientName.Text;
DateTime testDate = Convert.ToDateTime(RegDate.Value);
string staffID = lblStaffID.Text;
string doctorID = lblDoctorID.Text;
string hospitalName = lblSubscriberName.Text;

lib.insertWard_round_tbl(patientID, patientName, testDate, staffID, doctorID, hospitalName, ref msg);

if (msg != "")
{
    lblMessage.Text = "unable to update vital signs report as a result of " + msg;
    MessageBox.Show("Registration failed!!!");
    return;
}
MessageBox.Show("Successful!");
}
protected void Button3_Click(object sender, EventArgs e)
{
    if (txtPatientID.Text.Trim() == string.Empty)
    {
        MessageBox.Show("Enter the Patient name");
        return;
    }
}
}

```

Patient Form

```

public partial class Staff_Module_Patients_Form : System.Web.UI.Page
{
    SqlConnection conn = new SqlConnection("Data Source=.;Initial Catalog=HDBMS;Persist Security Info=True;User ID=sa;Password=yuiop123");

    protected void Page_Load(object sender, EventArgs e)
    {
        Panel9.Visible = true;
        Panel1.Visible = false;
        Panel2.Visible = false;
        Panel3.Visible = false;
        Panel4.Visible = false;
        Panel5.Visible = false;
        Panel7.Visible = false;
        Panel8.Visible = false;
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
        if (txtPatientID.Text.Trim() == string.Empty)
        {
            MessageBox.Show("Enter the Patient ID");
            return;
        }
        SqlDataAdapter adop = new SqlDataAdapter();
        SqlDataAdapter adip = new SqlDataAdapter();
        SqlDataAdapter adlp = new SqlDataAdapter();
        SqlDataAdapter adbp = new SqlDataAdapter();

        conn.Open();
        DataTable dt = new DataTable();
        DataTable dp = new DataTable();
        DataTable dl = new DataTable();
        DataTable db = new DataTable();
        adop = new SqlDataAdapter("select * from DOCTOR_REPORT_TBL where PATIENT_ID = " + txtPatientID.Text + "", conn);
        adip = new SqlDataAdapter("select * from VITAL_SIGNS_TBL where PATIENT_ID = " + txtPatientID.Text + "", conn);
        adlp = new SqlDataAdapter("select * from LAB_REPORT_TBL where PATIENT_ID = " + txtPatientID.Text + "", conn);
    }
}

```

```

adbp = new SqlDataAdapter("select * from PATIENT_PROFILE_BILLING_TBL where PATIENT_ID = '" + txtPatientID.Text + "'",
conn);
adop.Fill(dt);
UltraWebGrid3.DataSource = dt;
adip.Fill(dp);
UltraWebGrid2.DataSource = dp;
adlp.Fill(dl);
UltraWebGrid1.DataSource = dl;
adbp.Fill(db);
UltraWebGrid6.DataSource = db;
conn.Close();
try
{
    hdbmsClass lib = new hdbmsClass();
    DataSet ds = new DataSet();
    string msg = "";
    string patientID = txtPatientID.Text;
    hdbmsClass adp = new hdbmsClass();
    ds = lib.fetchPatientByID(patientID, ref msg);
    if (msg != "" || ds.Tables.Count == 0)
    {
        lblErrorMsg.Text = "unable to fetch patient as a result of " + msg;
        MessageBox.Show("Incorrect Details!!!");
        return;
    }
    lblPatientName.Text = ds.Tables[0].Rows[0]["PATIENT_NAME"].ToString();
    lblPatientID.Text = ds.Tables[0].Rows[0]["PATIENT_ID"].ToString();
    lblGender.Text = ds.Tables[0].Rows[0]["GENDER"].ToString();
    lblPhoneNumber.Text = ds.Tables[0].Rows[0]["MOBILE_NUMBER"].ToString();
    lblEmailAddress.Text = ds.Tables[0].Rows[0]["EMAIL_ADDRESS"].ToString();
    lblBloodGroup.Text = ds.Tables[0].Rows[0]["BLOOD_GROUP"].ToString();
    imgPassport.ImageUrl = ds.Tables[0].Rows[0]["IMAGE_UPLOAD"].ToString();
    if (checkBoxCertify.Checked == true)
    {
        Panel9.Visible = false;
        Panel1.Visible = true;
        Panel2.Visible = true;
        Panel3.Visible = true;
        Panel4.Visible = true;
        Panel5.Visible = true;
        Panel7.Visible = true;
        Panel8.Visible = true;
    }
}
catch
{
    MessageBox.Show("Incorrect Details!!!");
}
}

```

Reference Form

```

public partial class Reference_Forms_Category_Ref_Form : System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        hdbmsClass lib = new hdbmsClass();
        string msg = "";
        string categoryName = txtCategory.Text;

        lib.insertCategory_tbl(categoryName, ref msg);

        if (msg != "")
        {
            lbl.Text = "unable to update category table as a result of " + msg;
            MessageBox.Show("Registration failed!!!");
        }
    }
}

```

```

        return;
    }
    MessageBox.Show("Successful!");
    txtCategory.Text = "";
}
protected void Button2_Click(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    string msg = "";
    string categoryName = dropCategory.Text;
    string SubCategoryName = txtSubCategory.Text;

    lib.insertCategory_sub_tbl(categoryName, SubCategoryName, ref msg);

    if (msg != "")
    {
        lbl.Text = "unable to update sub category table as a result of " + msg;
        MessageBox.Show("Registration failed!!!");
        return;
    }
    MessageBox.Show("Successful!");
    txtSubCategory.Text = "";
}
protected void Button3_Click(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    string msg = "";
    string categoryName = dropCategory2.Text;
    string SubCategoryName = dropSubCategory.Text;
    string subSubCategoryName = txtSubSubCategory.Text;
    lib.insertCategory_sub_sub_tbl(categoryName, SubCategoryName, subSubCategoryName, ref msg);
    if (msg != "")
    {
        lbl.Text = "unable to update sub sub category table as a result of " + msg;
        MessageBox.Show("Registration failed!!!");
        return;
    }
    MessageBox.Show("Successful!");
    txtSubSubCategory.Text = "";
}
protected void dropCategory2_SelectedIndexChanged1(object sender, EventArgs e)
{
    hdbmsClass lib = new hdbmsClass();
    DataSet ds = new DataSet();
    string msg = "";
    string category = dropCategory2.SelectedItem.ToString();
    ds = lib.fetchCategorySubByCategory(category, ref msg);
    if (msg != "")
    {
        lbl.Text = "unable to fetch category as a result of " + msg;
        MessageBox.Show("Error fetching treatment!!!");
        return;
    }
    dropSubCategory.DataSource = ds;
    dropSubCategory.DataTextField = "SUB_CATEGORY_NAME";
    dropSubCategory.DataValueField = "SN";
    dropSubCategory.Items[0].Value = "Select";
}
}

```

APPENDIX II

LETTER OF INTRODUCTION TO HOSPITALS

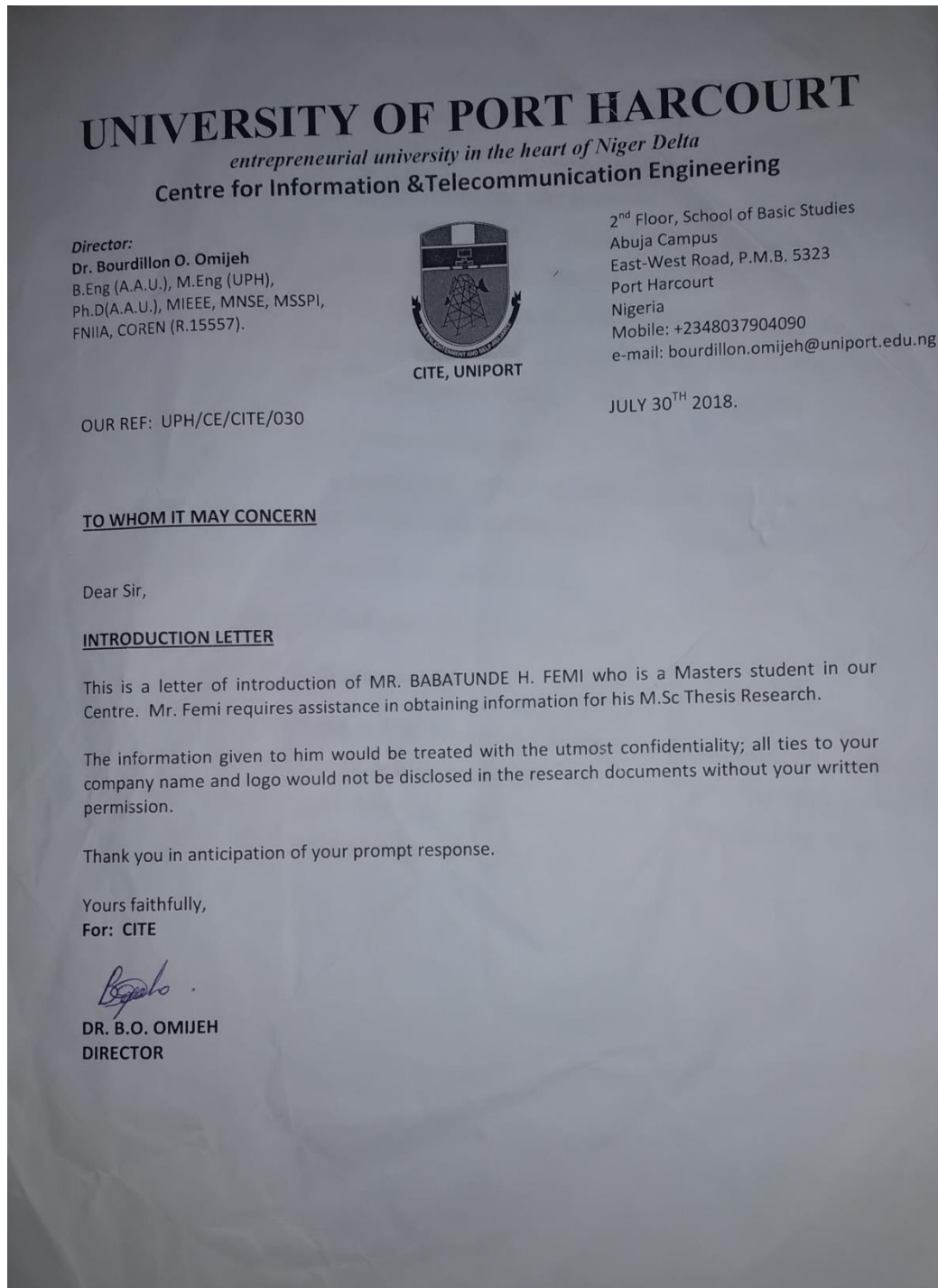


Fig 7.1 Letter of Introduction